A New Class of Location Measures, the Guard Estimator and Connections to Multiscale Variance Stabilization

G. P. Nason*, School of Mathematics, University of Bristol, United Kingdom

September 2012

Abstract

This article introduces a new class of location estimators by applying an existing measure of location to a set of multiscale means computed on the order statistics of a data set. Class members are shown to have interesting and desirable properties such as: the member using the geometric mean is always sandwiched between the geometric and arithmetic means. The member using the median reduces to a simple form, the guard estimator, which is the median of three quantities: the sample mean and two guard values based on two or four order statistics nearest the median. The guard estimator is unbiased, consistent (at least for symmetric distributions), computable in linear time, does not require tuning parameters and simulations suggest that it achieves high efficiency: almost matching the mean or median's better efficiency under different distributions. Guard's finite sample breakdown point demonstrates that it is highly robust even for small samples and matches the median's breakdown value asymptotically. Two examples exhibit the new location measures in action: one provides confirmation of a robust approach to establishing whether Shoshoni leather goods were designed to the 'Golden Ratio' standard; the other compares four functional measures of location for the Aberporth wind speed series.

The new class of location estimators is inspired by the member that uses the geometric mean which arises naturally from a theoretical analysis of multiscale variance stabilization (MVS) techniques. The article introduces maximum likelihood approaches for MVS techniques for independently and identically distributed data and sheds theoretical light on MVS by presenting analytical formulae for their Jacobians, a key component of the likelihood. The MVS techniques are shown empirically to compare favourably to the well-known Box-Cox transform, but do not dominate it.

KEYWORDS: LOCATION MEASURE, GUARD ESTIMATOR, MULTIMEANS, GEOMETRIC MEAN, MEDIAN, BREAKDOWN, VARIANCE STABILIZATION, HAAR-FISZ, HAAR WAVELET

1 Introduction

Given a set of data, $\mathbf{X} = \{X_1, \ldots, X_n\}$ for some integer $n > 0$ many people turn to the sample mean, $\bar{X} = n^{-1} \sum_{i=1}^{n} X_i$, to acquire a sense of the central tendency of the

*g.p.nason@bristol.ac.uk

data. The sample mean is natural, rapidly computable, widely understood as well as giving an accurate and efficient measure of the central tendency for many sets of data. It has been long known that, for many other sets of data, such as those contaminated with outliers, that the mean is often not so useful and a robust estimator, such as the median, is a more appropriate alternative. Measures of location and robustness are a fundamental area of study within statistics and there is a truly enormous literature see Hettmansperger and McKean (1998) or Huber and Ronchetti (2009) for comprehensive treatments.

This article introduces a new class of location estimators which are functions of order statistics. The general construction of the new class is via multiscale Haar father wavelet coefficients (multiscale means) but one member, the guard estimator, which we study in detail, reduces to an intriguing form seemingly unconnected to multiscale entities. Overall, our aim is to obtain estimators which are statistically efficient, robust, rapidly computable (linear computation time) and do not require tuning parameters.

The general class is introduced later, but by way of introduction, we now present one face of the guard estimator. The guard estimator takes advantage of the sample mean when it is 'well-behaved' but defers to statistics close to the median when it is not and is defined as follows.

Let $\{X_{(i)} : i = 1, \ldots, n\}$ be the usual (ascending) order statistics of $\mathbf{X}$. Then the estimator $\mathrm{guard}(\mathbf{X})$, for even n, is defined to be:

$$\mathrm{guard}(\mathbf{X}) = \mathrm{median}\left(\bar{X}_{[(n/2-1):n/2]}, \bar{X}, \bar{X}_{[(n/2+1):(n/2+2)]}\right), \tag{1}$$

where $\bar{X}_{[a:b]} = (b-a+1)^{-1}\sum_{i=a}^{b} X_{(i)}$ for $a, b \in \mathbb{N}$ and $a < b$. The guard estimator works by returning the sample mean when it is 'well-behaved', that is, between the *guard* values of $\bar{X}_{[(n/2-1):n/2]}$ and $\bar{X}_{[(n/2+1):(n/2+2)]}$. However, if outlying data pull the mean to outside of the guard values, then one or other of the guard values is returned instead. Since the guard values are close to the median the overall estimator is robust when the sample mean is displaced by outliers. The guard estimator for odd n can be constructed by replacing the two guard values by the order statistics $X_{((n-1)/2)}$ and $X_{((n+3)/2)}$ respectively.

Example 1 *Suppose the sample* $\mathbf{X} = (13, 10, 7, 9, 11, 8, 8, 7)$. *The usual sample mean* $\bar{X} = 9.125$. *The sorted sample is* $(7, 7, 8, 8, 9, 10, 11, 13)$. *To compute* $\mathrm{guard}(\mathbf{X})$ *note that* n *is even, so we require the following order statistics:* $X_{(3)} = X_{(4)} = 8$ *and* $X_{(5)} = 9, X_{(6)} = 10$. *Hence* $\bar{X}_{[3:4]} = 8$ *and* $\bar{X}_{[5:6]} = 9.5$. *Then*

$$\mathrm{guard}(\mathbf{X}) = \mathrm{median}(8, 9.125, 9.5) = 9.125.$$

In this case the mean is 'well-behaved' and its value is chosen for the final estimate. To demonstrate robustness: suppose now that the largest value of the sample, 13, is an outlier, 100, say. Clearly, the two guard values remain unchanged. However, the sample mean becomes $\bar{X} = 20$. *In this case* $\mathrm{guard}(\mathbf{X}) = \mathrm{median}(8, 20, 9.5) = 9.5$ *and the upper guard value is returned.*

Although our later methodology involves multiscale entities there is nothing inherently multiscale about guard. Fundamentally, this article is about combining existing measures of location. Guard combines the mean, median and two Walsh averages of order statistics close to the median.

As will be shown below guard is efficient, robust, can be computed in $\mathcal{O}(n)$ operations and does not require tuning parameters. In practice, the statistical properties of guard are reminiscent of the well-known Hodges-Lehmann (H-L) estimator of location, although H-L requires $\mathcal{O}(n^2)$ operations to compute, which is not 'fast'. Guard also possesses excellent robustness properties (that the mean lacks) and intriguing efficiency properties, as demonstrated in Section 5.2.

Before we delve into the details and properties of the particular guard estimator we first explain the structure of our paper. Part I of the paper is concerned with our new class of measures of location: the general class is introduced in Section 2 for dyadic length data. These new estimators work by applying an existing measure of location, such as the median, to the collection of Haar father coefficients of a set of (usually sorted) data. Section 3 describes some properties of the new class for unsorted data. Section 4 explains problems that occur arising from the use of unsorted data and introduces new measures based on the order statistics. Section 5 investigates properties of the 'sorted' measures and focuses attention on the guard estimator including proving equivalence between guard given in (1) and its multiscale face, its statistical properties including consistency and a theoretical result which establishes its excellent breakdown point. Section 6 extends the measures defined in Sections 3 and 5 for all $n > 0$, although guard, defined by (1), is already valid for all n. Further discussion and avenues for exploration appear in Section 7 and two examples in Section 8.

Part II of the paper sheds new theoretical light on multiscale variance stabilization (MVS): providing theoretical understanding of how they operate for iid data. Section 9 reviews the existing Box-Cox and MVS transforms and introduces modified versions of the multiscale ones for the iid data setup. Section 10 reviews the likelihood problem for stabilization, reviews it for Box-Cox and establishes a new result on the MVS transforms' Jacobians, and concludes with an empirical comparison of the different methods. The theoretical result on the Jacobians leads directly to the new class of location measures explored in Part I.

Part I: A New Class of Location Measures

2 Multiscale Measures of Location

Our new estimators depend on the data via a set of multiscale means (Haar father wavelet coefficients) which we will define next.

2.1 Multiscale Means

Definition 1 *(Multiscale Means) Let $J \in \mathbb{N}$, define $n = 2^J$ and suppose we have data $\mathbf{X} = \{X_i : X_i \in \mathbb{R}\}_{i=1}^n$. Set initial coefficients $c_{J,i-1} = X_i$ for $i = 1, \ldots, n$. Then perform the recursive operation:*

$$c_{j,i} = (c_{j+1,2i} + c_{j+1,2i+1})/2. \tag{2}$$

for $j = J-1, \ldots, 0$ and $i = 0, \ldots, 2^j - 1$. For $j = 0, \ldots, J-1$ and $i = 0, \ldots, 2^j - 1$ the quantity $c_{j,i}$ is called a **multiscale mean**. *Denote the set of all multiscale means of data $\mathbf{X}$ by $\mathcal{C}(\mathbf{X}) = \{c_{j,i} : j = 0, \ldots, J-1, i = 0, \ldots, 2^j - 1\}$.*

Remark 1 *$\mathcal{C}(\mathbf{X})$ is just the set of all $n-1$ discrete Haar father wavelet coefficients. A full definition of the discrete Haar wavelet transform is given in section 9.2.1 where we will use both father and mother coefficients for variance stabilization purposes.*

Remark 2 *For example, for $n = 8$, say, $c_{2,0} = (X_1 + X_2)/2$ is the sample mean of the first two observations, $c_{2,1} = (X_3 + X_4)/2$, and so on, and $c_{1,0} = (X_1 + X_2 + X_3 + X_4)/4$ and so on and $c_{0,0} = \bar{X}$ the overall sample mean. In other words, the $c_{j,i}$ are all sample means of subsamples of consecutive observations of dyadic lengths.*

Remark 3 *The order of the X_i within $\mathbf{X}$ matters. The set $\mathcal{C}(p(\mathbf{X}))$ is not necessarily equal to the set $\mathcal{C}(\mathbf{X})$, where p is a permutation of size n. We will say more on this in Section 4.*

2.2 New Multiscale Measures of Location

We begin straightaway by defining our new measure of location for sample sizes of length 2^J, we will extend this to general n later. In the following the set $\mathcal{D}$ is the domain of the data. For example, and usually, $\mathcal{D} = \mathbb{R}$ or $\mathcal{D} = \{x \in \mathbb{R} : x > 0\}$.

Definition 2 *Let $J \in \mathbb{N}$, define $n = 2^J$. Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a data set where each $X_i \in \mathcal{D}$. Let $m : \mathcal{D}^{n-1} \to \mathbb{R}$ be some measure of location. Then the* **(unsorted) multiscale measure of location** *is defined by*

$$\mathrm{umm}_m(\mathbf{X}) = m\{\mathcal{C}(\mathbf{X})\}. \tag{3}$$

We refer to the general class as (unsorted) multimeans.

Example 2 *Let $m =$ mean, be the usual sample mean and $\mathcal{D} = \mathbb{R}$. Here $\mathrm{umm}_{\mathrm{mean}}(\mathbf{X})$ is the sample mean of all the Haar father wavelet coefficients of $\mathbf{X}$. Consider a fixed level $j = 0, \ldots, J$ then $\sum_{i=0}^{2^j-1} c_{j,i} = (\sum_{i=1}^{n} X_i)/2^{J-j} = n\bar{X}/2^{J-j} = 2^j\bar{X}$. Then:*

$$\mathrm{umm}_{\mathrm{mean}}(\mathbf{X}) = (n-1)^{-1} \sum_{j=0}^{J-1} \sum_{i=0}^{2^j-1} c_{j,i} = (n-1)^{-1}\bar{X} \sum_{j=0}^{J-1} 2^j = \bar{X}.$$

In other words, $\mathrm{umm}_{\mathrm{mean}}(\mathbf{X})$ is just the regular sample mean of $\mathbf{X}$.

Example 3 *m is the sample median and $\mathcal{D} = \mathbb{R}$. Later we will show how a version of this estimator underlies the* guard *estimator given in* (1)*.*

Example 4 *m is the geometric mean and $\mathcal{D} = \{x \in \mathbb{R} : x > 0\}$. We denote this estimator $\mathrm{umm}_{\mathrm{GM}}$ and consider some of its properties below.*

Example 5 (Gaussian) *The following is an independent sample of four from $N(5, 1)$: $x_1 = 3.187594, x_2 = 4.615795, x_3 = 3.790022, x_4 = 5.300464$. Note that all the x_i are non-negative and so it is possible to compute the geometric mean and its multimean version. Denote the geometric mean of a set of data $\mathbf{X}$ by $\mathrm{GM}(\mathbf{X})$. For this data, to 3 decimal places, $\bar{x} = 4.223 = \mathrm{umm}_{\mathrm{mean}} = \mathrm{umm}_{\mathrm{median}}$, $\mathrm{GM} = 4.146$, $\mathrm{median} = 4.203$ and $\mathrm{umm}_{\mathrm{GM}} = 4.215$.*

Example 6 (Poisson) *The following is an independent sample of eight observations from* $\mathrm{Poiss}(10)$*:* $13, 10, 7, 9, 11, 8, 8$ *and* 7*. Here* $\bar{x} = \mathrm{umm}_{\mathrm{mean}} = \mathrm{umm}_{\mathrm{median}} = 9.125$*,* $\mathrm{GM} = 8.928$*,* $\mathrm{median} = 8.5$ *and* $\mathrm{umm}_{\mathrm{GM}} = 9.046$*, to three decimal places.*

Note, in general, $\mathrm{umm}_{\mathrm{mean}} \neq \mathrm{umm}_{\mathrm{median}}$.

3 Properties of the (unsorted) Multimeans

3.1 An Inequality.

It is well known that $\mathrm{GM}(\mathbf{X}) \leq \bar{X}$ with equality iff $X_1 = \cdots = X_n$. The next result shows that $\mathrm{umm}_{\mathrm{GM}}$ slots nicely between $\mathrm{GM}(\mathbf{X})$ and $\bar{X}$ for all data sets for which $X_i > 0$.

Theorem 1 *Let* $X_1, \ldots, X_n$ *be a data set, where* $n = 2^J$*,* $J \in \mathbb{N}$ *and* $X_i \geq 0$ *for all* $i = 1, \ldots, n$*. Let* $\mathrm{umm}_{\mathrm{GM}}(\mathbf{X})$ *be defined as in Example 4. Then:*

$$\mathrm{GM}(\mathbf{X}) \leq \mathrm{umm}_{\mathrm{GM}}(\mathbf{X}) \leq \bar{X}, \tag{4}$$

for all $\mathbf{X}$ *with equality iff* $X_1 = \cdots = X_n$*.*

The proof of Theorem 1 appears in Appendix A. The result is general and holds irrespective of the distribution or ordering of the entries of $\mathbf{X}$.

Remark 4 *The geometric mean tends to be used for data sets that are 'multiplicative' in nature, or where different parts of the sample vary in scale. For example, the log-normal distribution is often cited as a typical distribution where the geometric mean can be used and for which it is the maximum likelihood estimator of the location parameter.* AM *has wider application. However, for data sets where the underlying distribution is not known then this (unsorted) multimean (or rather the sorted version below) might be an attractive complement.*

Remark 5 *The* $\mathrm{umm}_{\mathrm{GM}}$ *estimator computes arithmetic means (to form the* $\mathcal{C}$*) and computes their geometric mean. So,* $\mathrm{umm}_{\mathrm{GM}}$ *uses both arithmetic and geometric means in its formation and the result is an estimator that lies between them. This behaviour, where the combined estimator 'lies between' the two components is reoccurring theme in this article: for example, in Figures 1, 2 and the example in Section 8.2.*

Remark 6 *The quantity* $\mathrm{umm}_{\mathrm{GM}}$ *is a combination of means of the data and, in practice, as it can be less extreme than either, under different circumstances. For example, if a single* $X_i \to 0$ *then the geometric mean can get dragged to zero, but the* $\mathrm{umm}_{\mathrm{GM}}$ *will not be. More will be said on robustness in Section 5.*

3.2 Equivariance to Affine Transformations of the Data

Any $c_{j,i} \in \mathcal{C}(\mathbf{X})$ is equivariant with respect to affine transformations. In other words if $Y_i = aX_i + b$ then the Haar father coefficients of Y_i are $ac_{j,i} + b$, due to equivariance inheritance from the mean. Hence, our new estimators inherit the equivariance properties of the location statistic, m, in Definition 2.

In particular, neither GM nor $\mathrm{umm}_{\mathrm{GM}}$ are equivariant with respect to translations, i.e. $\mathrm{umm}_{\mathrm{GM}}(\mathbf{X}+b) \neq \mathrm{umm}_{\mathrm{GM}}(\mathbf{X}) + b$ for $b \in \mathbb{R}$. However, computational evidence suggests, and we conjecture that, the 'geometric mean' (unsorted) multimean is "more equivariant" with respect to translation than the geometric mean itself in that if one defines $\mathrm{Eq}_{\mathrm{GM},\mathbf{X}}(b) = \{\mathrm{GM}(\mathbf{X}+b) - \mathrm{GM}(\mathbf{X})\} \,/\, \mathrm{GM}(\mathbf{X})$ then we conjecture $\mathrm{Eq}_{\mathrm{umm}_{\mathrm{GM}},\mathbf{X},b} \leq \mathrm{Eq}_{\mathrm{GM},\mathbf{X},b}$ for all data sets $\mathbf{X}$ and positive $b \in \mathbb{R}$.

4 Invariance to Data Order

The astute reader will have noticed that the (unsorted) multimeans are not invariant to the ordering of the X_i, because discrete wavelet transforms are not invariant and neither are the $\{c_{j,i}\}$. Multimeans *are* invariant to some reorderings of the data, e.g. swapping any X_{2i-1} with X_{2i} for $i = 1, \ldots, n/2$ does not change the values of any of the father wavelet coefficients, $c_{j,k}$, and the (unsorted) multimean of such swapped data remains the same.

It seems highly undesirable for a location measure not to be invariant to the order of the data. Several remarks can be made about this situation.

1. (How many different unique values are there?) Let P_n denote the number of possible different values of $\mathrm{umm}(\mathbf{X}_p)$ where $\mathbf{X}_p$ are the data ordered according to the permutation $p \in \mathcal{P}$, where $\mathcal{P}$ is the set of all possible permutations of $(1, \ldots, n)$. Numerically P_n appears to increase rapidly, i.e. $P_2 = 1, P_4 = 3, P_8 = 315$. An interesting open question is to derive a formula for P_n.

2. (How different are the actual values?) Let $R_p = \mathrm{umm}_{\mathrm{GM}}(\mathbf{X}_p)$, e.g. For a given data set $\mathbf{X}$ let $V(\mathbf{X})$ be the variance over the different values of R_p over all $p \in \mathcal{P}$. For example, we estimated $V(\mathbf{X})$ empirically over ten iid samples of length 8 from $N(10, 1)$. The mean of the variance over the ten samples was 4.53×10^{-5}. We observed similar behaviour in other examples. For this simulation the spread of possible values of R_p seems to be extremely small. Clearly, a mathematically precise answer to this question would be desirable.

3. (Mitigation?) One possibility might be to compute $(n!)^{-1} \sum_{p \in \mathcal{P}} \mathrm{umm}_{\mathrm{GM}}(\mathbf{X}_p)$ the average of $\mathrm{umm}_{\mathrm{GM}}(\mathbf{X}_p)$ over all permutations, but this is likely to be highly impractical even for fairly small n. A more computationally realistic possibility is to compute $\mathrm{umm}(\mathbf{X}_p)$ over all n cyclic shifts of the data in $\mathbf{X}$, which can be computed in $\mathcal{O}(n \log n)$ operations using the nondecimated wavelet transform, see Coifman and Donoho (1995); Nason and Silverman (1995). However, this latter possibility is also not invariant to all permutations.

Another approach to solving the problem of the lack of invariance was proposed by Motakis et al. (2006) where the data arose as the result of a blocked experiment: the data could be permuted within the block and not affect any conclusions from the experiment. Motakis et al. (2006) suggested sorting the data before analysis by a wavelet-based method. Their rationale was that sorting meant that the (mother) wavelet coefficients could not be arbitrarily large solely because of the data order and that sorting conferred a type of sparsity on the coefficients.

We adopt this approach here: sort the data and then apply the multimean.

Definition 3 *Let $J \in \mathbb{N}$, define $n = 2^J$. Let $\mathbf{X}_{p^*} = (X_{(1)}, \ldots, X_{(n)})$ be the order statistics of $\mathbf{X}$ where each $X_i \in \mathcal{D}$. Here p^* is the permutation that sorts $\mathbf{X}$ into ascending order. Let $m : \mathcal{D}^{n-1} \to \mathbb{R}$ be some measure of location. Then the* **(sorted) multiscale measure of location** *is defined by:*

$$\mathrm{mm}_m(\mathbf{X}) = m\{\mathcal{C}(\mathbf{X}_{p^*})\}. \tag{5}$$

We refer to the general class as sorted multimeans, or just multimeans. Note, for $n = 2$ then the set $\mathcal{C}$ only contains one element $(X_1 + X_2)/2$.

Example 7 (Gaussian revisited) *Using the $N(5, 1)$ sample of four from Example 5 we have $\mathrm{mm}_{\mathrm{GM}} = 4.180$ and $\mathrm{mm}_{\mathrm{median}} = 4.223$ to three d.p.*

Example 8 (Poisson revisited) *Using the $\mathrm{Poiss}(10)$ sample of eight from Example 6 we have $\mathrm{mm}_{\mathrm{GM}} = 8.977$ and $\mathrm{mm}_{\mathrm{median}} = 9.125$ to three d.p.*

Remark 7 *The well-known L-estimators are linear combinations of order statistics or, more generally, a linear combination of some univariate function of each of them, see Huber and Ronchetti (2009). Our sorted multimeans introduced in Definition 3 are not L-estimators but they are more general functions of order statistics.*

5 Properties of the (sorted) Multimeans

5.1 The minimum of $\mathrm{umm}_{\mathrm{GM}}$ is $\mathrm{mm}_{\mathrm{GM}}$

It turns out that the geometric mean multimean of the sorted data is equal to the minimum of the unsorted version over all permutations $p \in \mathcal{P}$.

Proposition 1 *Let $J \in \mathbb{N}$, define $n = 2^J$. Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a data set where each $X_i > 0$. Then*

$$\mathrm{mm}_{\mathrm{GM}}(\mathbf{X}) = \min_{p \in \mathcal{P}} \mathrm{umm}_{\mathrm{GM}}(\mathbf{X}_p). \tag{6}$$

The proof is in the appendix.

Remark 8 *Note that the permutation p^* that sorts $\mathbf{X}$ achieves the minimum, but it is not necessarily the only permutation that does so.*

Remark 9 *Proposition 1 is not valid for $\mathrm{mm}_{\mathrm{median}}$. It is valid trivially for $\mathrm{mm}_{\mathrm{mean}}$ as $\mathrm{umm}_{\mathrm{mean}}$ is invariant over $\mathcal{P}$ as shown by Example 2.*

5.2 The guard estimator

The following result explains the origin of the guard estimator introduced in (1).

Theorem 2 *Let $J \in \mathbb{N}$, define $n = 2^J$. Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a data set where each $X_i \in \mathbb{R}$. Then*

$$\mathrm{guard}(\mathbf{X}) = \mathrm{mm}_{\mathrm{median}}(\mathbf{X}). \tag{7}$$

The proof is in the appendix. It is curious to see that the median of a set of multi-scale means, $\mathcal{C}(\mathbf{X})$, reduces to the median of three statistics: the sample mean and the two guard values, which are not inherently multiscale. The dual representation of $\mathrm{guard}(\mathbf{X})$ is useful: below we use the 'median-of-three' representation in (1) to show unbiasedness and consistency; we use the 'median of the set $\mathcal{C}$' to determine its robustness properties.

The guard estimator is unbiased for the mean and mean-square consistent under the assumptions of the following result.

Proposition 2 *(Statistical Properties). Let $J \in \mathbb{N}$, define $n = 2^J$. Let $\mathbf{X} = (X_1, \ldots, X_n)$, $X_i \in \mathbb{R}$ and each X_i arising from the absolutely continuous distribution, F, with continuous density f symmetric about μ, also the (finite) mean which exists. Then $\mathrm{guard}(\mathbf{X})$ is unbiased for μ and is (mean-square) consistent.*

The proof is in the appendix. The rate of convergence for the mean squared error is the standard parametric rate n^{-1}.

Figures 1 and 2 depict empirical variances of the estimators for four different underlying distributions: standard normal, Cauchy, double exponential (standard Laplace) and the Uniform distribution on $[-0.5, 0.5]$. Where the mean exists (all apart from Cauchy) the guard estimator's variance is always between that of the mean and the median, the same can be said for the Hodges-Lehmann estimator. For both guard and Hodges-Lehmann, neither obtain the best rates of mean or median for standard normal/uniform and Cauchy/double exponential respectively, but both perform better than the mean/median when they perform less well.

Figures 3 and 4 show the variance in more detail as a function of sample size for standard normal and double exponential distributions. These two figures verify established wisdom that: for normal data the mean is more efficient than the median and for double exponential data the situation is reversed, see Arnold et al. (1992) page 225, e.g. However, remarkably, the guard estimator, has efficiency close to the mean for standard normal data and efficiency close to the median for the double exponential data, at for small-moderate n. Hence, for these examples at least, guard appears to be using the mean when it is most useful and the guard values, very close to the median when the mean is less useful.

5.3 Breakdown point of the guard estimator

Huber and Ronchetti (2009, Chapter 11) discuss the breakdown point: "roughly, the smallest amount of contamination that may cause an estimator to take on arbitrarily large aberrant values". Further, (Section 11.1), they remark "The examples show that for small samples (say $n = 10$ or so) a high breakdown point (larger than 25%) is desirable to safeguard against unavoidable random asymmetries involving a small number of aberrant observations." This leads us onto consideration of the breakdown point for the guard estimator.

Theorem 3 *(Breakdown Point) The finite sample breakdown point for the $\mathrm{guard}(\mathbf{X})$ estimator is $\frac{1}{2} - \frac{1}{n}$ for $n = 2^J \geq 4$. The asymptotic breakdown point is $\frac{1}{2}$.*

The proof is in the appendix. So even for small $n \geq 4$ the breakdown point is $\geq \frac{1}{4}$ (e.g. the breakdown for sample sizes of $n = 4, 8$ is $\frac{1}{4}$ and $\frac{3}{8}$. For moderate length

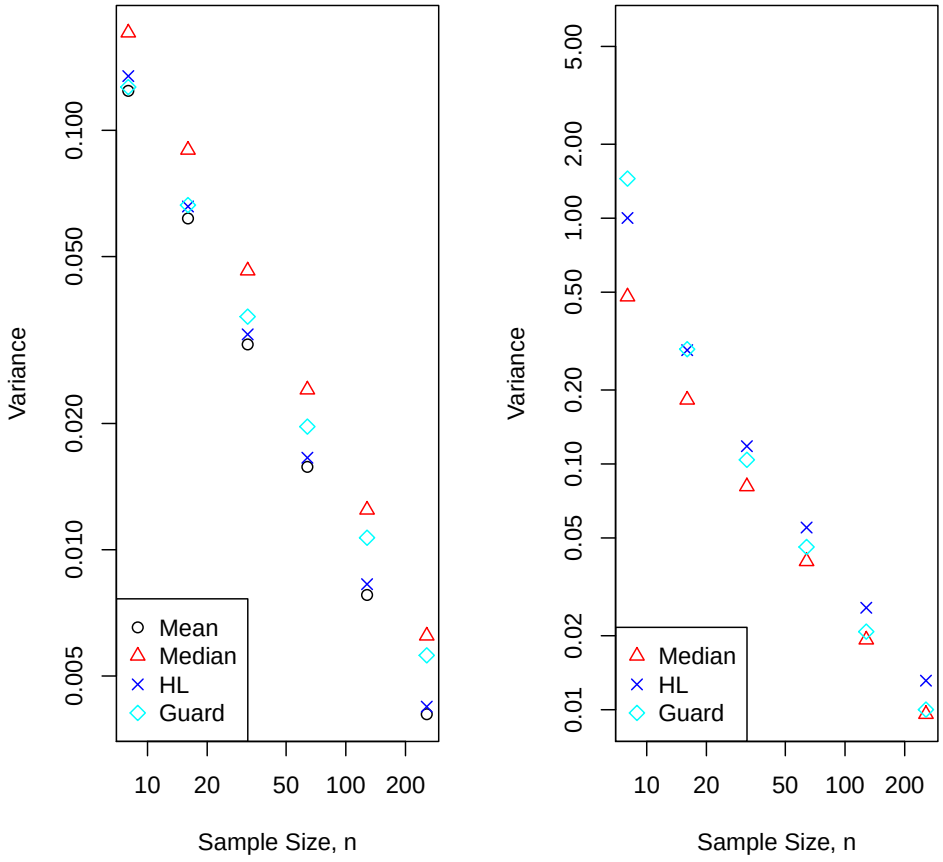

Figure 1: Empirical variance of four estimators in estimating location parameters versus sample size for iid samples from: *Left:* standard normal; *Right:* Cauchy distribution. Note: the mean estimator is omitted from the right-hand plot as the mean does not exist for Cauchy (and the estimator performs badly, as expected). (Monte Carlo estimates over 10000 simulations).

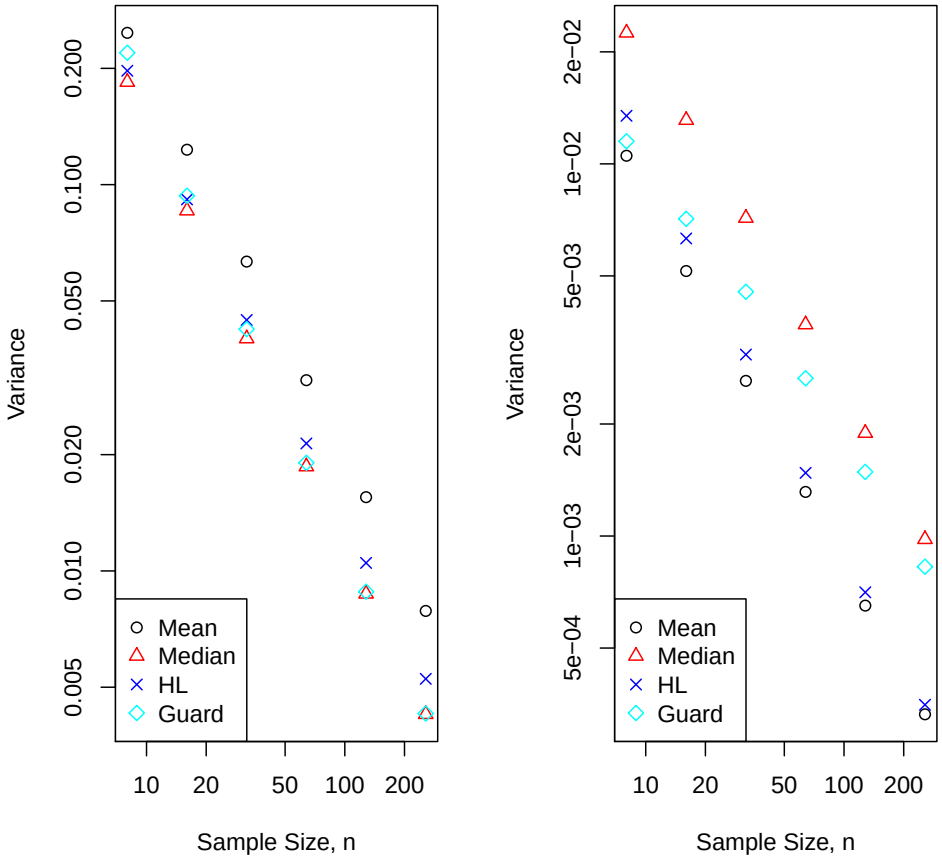

Figure 2: Empirical variance of four estimators in estimating location parameters versus sample size for iid samples from: *Left:* double exponential; *Right:* Uniform distribution on $[-0.5, 0.5]$. (Monte Carlo estimates over 10000 simulations).

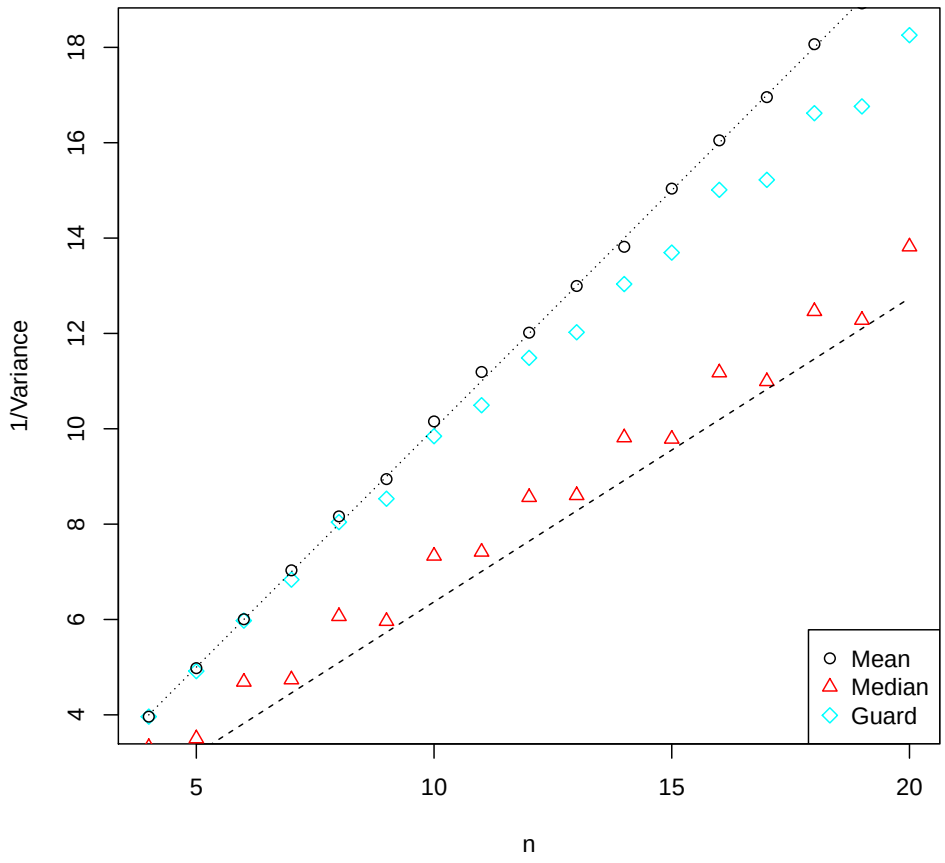

Figure 3: Inverse variance versus sample size, n, for the guard, mean and median estimators computed on standard normal data (over 20000 simulations). The upper dotted line corresponds to variance $= n^{-1}$, the lower dashed line corresponds to variance $\approx 1.57n^{-1}$ the asymptotic rates for the mean and median respectively. Note: 'better' estimator performance results in inverse variance *higher* up the vertical axis.

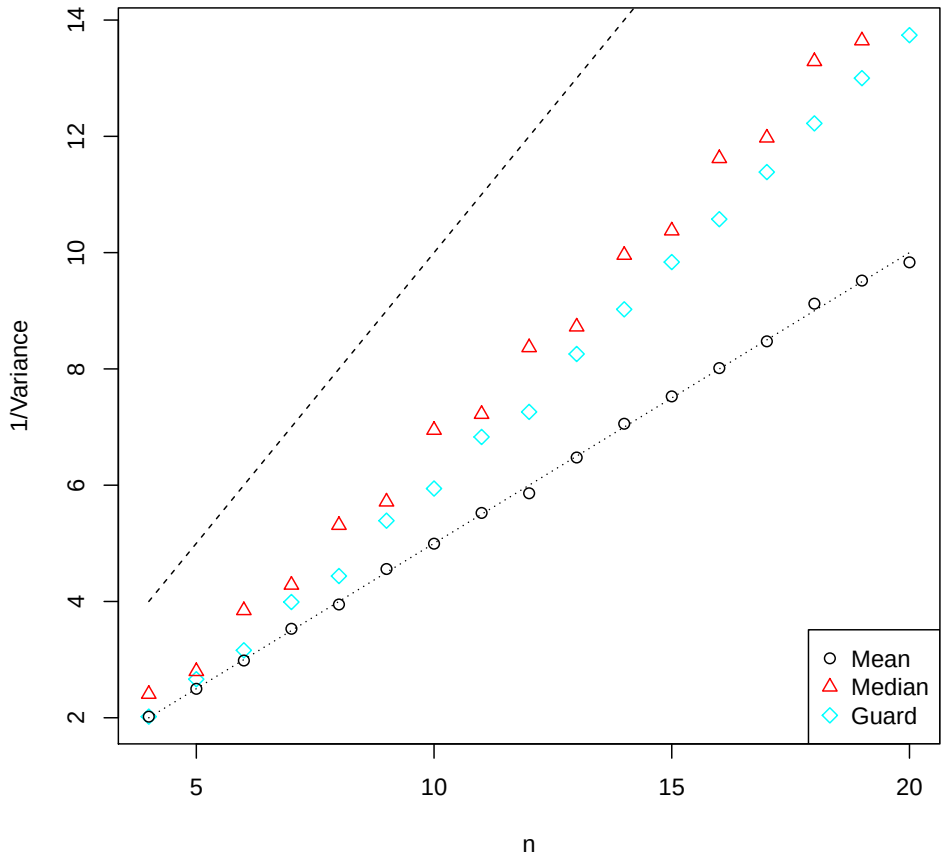

Figure 4: Inverse variance versus sample size, n, for the guard, mean and median estimators computed on double exponential data (over 20000 simulations). The lower dotted line corresponds to variance $= 2n^{-1}$, the upper dashed line corresponds to variance $= n^{-1}$ the asymptotic rates for the mean and median respectively. Note: 'better' estimator performance results in inverse variance *higher* up the vertical axis.

samples the guard estimator is very robust, e.g. for $n = 64$ the breakdown point is approximately 0.48 and nearly as robust as the median. The guard estimator compares favourably to the Hodges-Lehmann estimator. Geyer (2006) and Huber and Ronchetti (2009) show that for $n = 8, 16, 32, 64$ the finite sample breakdown is $\frac{3}{8}, \frac{5}{16}, \frac{10}{32}, \frac{19}{64}$. In other words, the breakdown points for the guard and Hodges-Lehmann estimator is the same for $n = 8$, but thereon that of Hodges-Lehmann decreases and that of guard increases. Asymptotically, the breakdown point for Hodges-Lehmann is $1 - 1/\sqrt{2} \approx 0.29289$. Hence, the robustness properties of guard are attractive.

6 Multimeans for arbitrary n

6.1 Definition

Although the guard estimator in (1) is clearly valid for all $n \in \mathbb{N}$, definitions 2 and 3 for the unsorted and sort multimeans only cover the case for $n = 2^J$, which is clearly inadequate for everyday applications. However, the multimeans can be extended to arbitrary integers n by computing *every possible* father wavelet coefficient, $c_{j,k}$, and then taking the appropriate measure of location, m, of those. Let $\lfloor x \rfloor$ denote the largest integer less than x. To be precise revisit the definition of the father wavelet coefficients from Definition 1 by redefining the following, more general, multiscale means.

Definition 4 (Arbitrary n father wavelet coefficients) *Let $n \in \mathbb{N}$. Define $J = \lfloor \log_2 n \rfloor$. Let $\tilde{c}_{J,i-1} = X_i$ be a data set for $i = 1, \ldots n$. Denote the arbitrary n father wavelet coefficients by $\tilde{c}_{j,k}$ for $j = 0, \ldots, J-1$ and $k = 1, \ldots, n_j$ where n_j is the number of such coefficients at level j. For $m \in \mathbb{N}$ define $\tilde{m} := m$ if m is even and $\tilde{m} := m - 1$ if m is odd. Define coarser scale father wavelet coefficients from finer scale ones by*

$$\tilde{c}_{j-1,i} = (\tilde{c}_{j,2i} + \tilde{c}_{j,2i+1})/2, \tag{8}$$

for $i = 0, \ldots, (\tilde{n}_j - 1)/2$. Define $\tilde{n} = \sum_{j=0}^{J-1} n_j$.

Definition 5 (Multimeans for arbitrary n) *Simply replace $\mathcal{C}(\mathbf{X})$ (or the sorted version) by the new set of arbitrary n father wavelet coefficients from Definition 4.*

Note, that the multimeans for arbitrary n agree with those for dyadic n. However, it is no longer the case that $\mathrm{guard}(\mathbf{X}) = \mathrm{mm}_{\mathrm{median}}(\mathbf{X})$, nor is Theorem 1 valid for the arbitrary n case. An interesting question would be to investigate how far outside the bounds of Theorem 1 the multiscale mean for arbitrary n could be for a 'regular' set of data, e.g. one with no significant outliers.

Example 9 (Linear Data) *Suppose $X_i = i$ for $i = 1, \ldots, 10$. Here $n = n_3 = 10$ and so $J = \lfloor \log_2(10) \rfloor = 3$. The quantity $\tilde{n}_3 = 10$ since n_3 is even. Hence $c_{2,0} = 1.5, c_{2,1} = 3.5, c_{2,2} = 5.5, c_{2,3} = 7.5, c_{2,4} = 9.5$ with $n_2 = 5$. Hence, $\tilde{n}_2 = 4$ and so $c_{1,0} = 2.5$ and $c_{1,1} = 6.5$, $n_1 = 2 = \tilde{n}_1$ and finally $c_{0,0} = 4.5$. The geometric multimean of all the $c_{j,k}$ is $\mathrm{mm}_{\mathrm{GM}}(\mathbf{X}) = 4.44$ (to 2 d.p.), note the data are already sorted here.*

6.2 Computational Effort

As well as efficiency, accuracy and robustness it is important to consider the computational effort required to compute the various estimates. It is particularly important when comparing estimators. For example, two estimators might give comparable statistical performance, but one might be dramatically more computationally efficient than the other.

Computing regular arithmetic and geometric means can be performed in $\mathcal{O}(n)$ operations, as can the median, see Sedgewick and Wayne (2011). All the unsorted multimeans for $n = 2^J$ are also $\mathcal{O}(n)$ if their location function m is. This is because the discrete father wavelet coefficients can be computed using the fast $\mathcal{O}(n)$ pyramid algorithm of Mallat (1989). Producing the sorted multimeans requires the order statistics which can be achieved in $\mathcal{O}(n \log n)$ operations.

The guard estimator is special in that it can be computed in $\mathcal{O}(n)$ operations. This is because its constituent parts: the mean and the guard values can be computed in $\mathcal{O}(n)$ operations. The guard values rely on a small finite number (two for odd n, four for even n) of order statistics which can be computed by the same sort of selection algorithm for the median mentioned in Sedgewick and Wayne (2011). These few order statistics can be computed in R using the `partial` option to the `sort.int` function.

For the arbitrary n version the number of operations $N(n)$ for a data set of size n is given by the recursive formula: $N(1) = 0$ and $N(2n + 1) = N(2n)$, $N(2n) = n + N(n)$ which is effectively $\mathcal{O}(n \log n)$.

7 Discussion, Extension, Questions

Remark 10 *For the odd-n* guard *estimator the two guard values are $X_{((n-1)/2)}$ and $X_{((n+3)/2)}$ as introduced in Section 1. A less robust estimator could be achieved by moving the guard values further 'out', away from the median. For example, the next most robust estimator would be:* $\mathrm{median}(X_{((n-3)/2)}, \bar{X}, X_{((n+5)/2)})$*, and less robust estimators still could be obtained by moving the guards 'out' even further. This process is similar to forming trimmed measures of location. Similar comments apply to the even-n version.*

Remark 11 *The multimeans are functions of the multiscale means $\mathcal{C}$. One might ask what happens if the multiscale means are replaced by others measures of location. For example, rather than form the means of (X_1, X_2), (X_3, X_4) and (X_1, X_2, X_3, X_4) one forms, e.g. the median, or some other location measure.*

Remark 12 *It is interesting to speculate what would happen if we moved from dyadic multimeans to triadic multimeans, or higher orders. That is, we replace formulae like $(c_{j,2i} + c_{j,2i+1})/2$ in* (2) *by formulae such as $(c_{j,3i} + c_{j,3i+1} + c_{j,3i+2})/3$. Here, for the moment, we assume $n = 3^J$ but further investigation might yield a simple estimator for all integers n such as* guard*. Preliminary simulations indicate Theorem 1 may well hold. It is not clear whether the median of triadic coefficients has a representation similar to* (1) *for the* $\mathrm{guard}(\mathbf{X})$ *estimators.*

Remark 13 *Another variant might replace the Haar father wavelet coefficients, $\mathcal{C}$, by those associated with more general wavelets or related lifting schemes. Lifting is a*

generalised multiscale paradigm, generalising wavelets to domains such as irregularly-spaced multivariate data or graphs and could help in the construction of weighted means in those cases. See Jansen et al. (2009), for examples and a list of references.

Remark 14 *The* guard *estimator takes the median of* **three** *quantities: the sample mean and two other quantities that are essentially the closest they can be to the median, without being the median. Can we produce a comparable estimator by some function of* **two** *of the quantities, or with just the mean and the median? Obviously, taking the mean of the sample mean and the median will not work: it is not robust for one thing, and the median reduces to the same thing. So, maybe three quantities are required?*

Remark 15 *Is there a version of these new estimators for multivariate data?*

Remark 16 *This article has concentrated on measures of location. Are there versions for estimating scale? For example, define* $\mathrm{var}\,\mathbf{X}_{[a:b]}$ *by* $\mathrm{var}(X_a, \ldots, X_b)$*. Then compute the equivalent of the* $\mathcal{C}$ *by computing variance on every dyadic subsequence (e.g.* $\mathrm{var}(X_1, X_2)$*,* $\mathrm{var}(X_3, X_4)$*,* $\mathrm{var}(X_5, X_6)$ *and so on,* $\mathrm{var}(X_1, X_2, X_3, X_4)$*,* $\mathrm{var}(X_5, X_6, X_7, X_8)$ *and so on, and so on for coarser scales.) Then, one obtains a set of multiscale variances rather than a set of multiscale averages. The final estimator of scale applies some measure of location,* m*, to the set of multiscale variances. Such a procedure will not work if it is applied to the order statistics as the variance is then seriously underestimated (as the variances are akin to mother wavelet coefficients and the point of sorting for the mothers is to achieve the sparsest, or least variable, sequence). Preliminary numerical calculations using the mean for* m*, show not unreasonable results encouraging further exploration.*

8 Examples

8.1 Shoshoni Data

Hettmansperger and McKean (1998) present width-to-length ratios of beaded rectangles used by the Shoshoni Indians to decorate leather goods. There are 20 observations which are

```
0.553 0.570 0.576 0.601 0.606 0.606 0.609 0.611 0.615 0.628
0.654 0.662 0.668 0.670 0.672 0.690 0.693 0.749 0.844 0.933
```

An oft-asked question for this data is whether the ratios are consistent with the famous (inverse) Golden Ratio $\phi^{-1} = 2/(1+\sqrt{5}) \approx 0.618034$. Table 1 shows 90% confidence intervals constructed through bootstrap resampling of a selection of the empirical location measures described in this paper. Hettmansperger and McKean (1998) remark that the Shoshoni data appear to contain two outliers. As a consequence they put more faith in the 'HM Interp' or 'Median' based confidence intervals (which are robust and both confidence intervals contain ϕ^{-1}) compared to the arithmetic mean bootstrap or t-test which both indicate that the Shoshoni are not using the Golden Ratio standard. The outliers appear to be outliers on the log-scale also and hence the geometric mean might not be robust to them either (and its confidence interval does not cover ϕ^{-1}). The

Table 1: Point estimate of location and associated 90% bootstrap confidence intervals for the Shoshoni data (to 4 d.p., 10^7 simulations). HM Interp. corresponds to the interpolated L_1 confidence interval from Hettmansperger and McKean (1998) which has 'achieved confidence' of 90% .

Method	Point Est.	90% CI	CI width	Has ϕ^{-1}?
Arithmetic Mean, $\bar{X}$	0.6605	[0.6296, 0.6957]	0.0661	✗
Geometric Mean, GM	0.6550	[0.6270, 0.6875]	0.0605	✗
Hodges-Lehmann	0.6420	[0.6190, 0.6775]	0.0585	✗
Median	0.6410	[0.6090, 0.6700]	0.0610	✓
HM Interp.	0.6410	[0.6087, 0.6702]	0.0615	✓
Guard	0.6580	[0.6120, 0.6700]	0.0591	✓

Hodges-Lehmann statistic is more robust than the arithmetic mean and its confidence interval does not cover ϕ^{-1}, but only just misses.

Our new guard estimator has a confidence interval that does cover ϕ^{-1}, and we know guard to be more robust than Hodges-Lehmann, from section 5.3. The guard estimator's point estimate is roughly halfway between that of the mean and the median and, even though the length of its confidence interval is smaller than all apart from Hodges-Lehmann, its confidence interval still covers ϕ^{-1}.

8.2 Aberporth wind data

Figure 5 shows an 85 day hourly wind speed record taken at the Aberporth weather station during 1995. Wind is a key renewable energy source of increasing importance and for wind farm planning reasons it is often important to understand the typical wind speed characteristics of a potential site. Small differences in the assessment of typical behaviour can result in significant consequences: whether the site is developed or not, how much energy, and hence profit, can be extracted from the site in the long term. This data set was previously analyzed by Nason and Sapatinas (2002), Cardinali and Nason (2010) and Moura et al. (2012). We think of the time series in Figure 5 as hourly observations on a continuous time series $X(s)$. To obtain information on the 'typical day' we reconfigure $X(t)$ into a functional time series $\{X_k(t) : k \in \mathbb{N}, t \in (0, 24)\}$ where k indexes a day and t the continuous time, in hours, within that day and $X_k(t) = X(t + 24k)$, see, for example, Bosq (1991). For the wind data this results in $k = 1, \ldots, 85$ days each of 24 hours time duration.

Four measures of location for the functional wind series are shown in Figure 6. Functional measures of location were obtained by applying the mean, median, Hodges-Lehmann and guard estimators to the first five Fourier coefficients of each of the 85 series and then applying the inverse Fourier transform to the summary statistics in each case. The general pattern in Figure 6 seems to be slower winds at night and then building faster during the day, which is often observed in wind data, see Emeis (2001) for example. Broadly, the Hodge-Lehmann and mean estimates of location are similar, as are the median and the guard estimates. In terms of level the guard estimator seems not to be 'in between' the mean and the median. However, in terms of time shift, the peak of the mean, guard and median estimates is 14:58, 15:04 and 15:50 hours

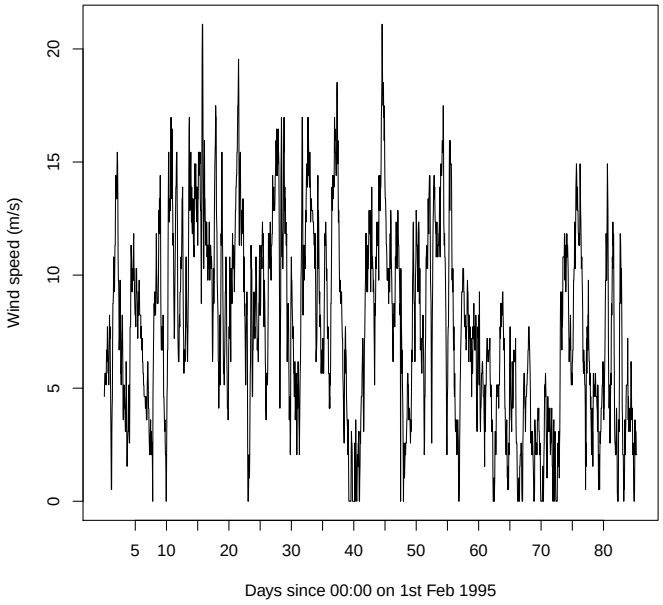

Figure 5: Hourly wind speeds at Aberporth Meteorological Office Station

respectively. So, roughly, the guard estimator's time shift is somewhere between the mean and median. This ordering is due to the measures of location acting on the series Fourier coefficients resulting in these phase orderings. A further interesting feature is the fact that the measures of location are all much closer to each other just before 10am, maybe reflecting increases in symmetry in the distributions of the Fourier coefficients around this time.

Part II: Variance Stabilization: Source of the Location Measures

Variance stabilization through transformation is a popular and commonly performed technique in statistics. For example, analysts routinely try log and/or square-root transformations to draw data towards homoscedasticity and/or normality. Even today, for some problems variance stabilization is important and essential. For example, in astrononomical image processing, where variance stabilization methods are used in combination with other methods to "recover important structures of various morphologies in (very) low-count images" and demonstrate that such techniques are "competitive relative to many existing denoising methods", Zhang et al. (2008).

This part of the article is concerned with three classes of variance stabilization which are defined in Section 9. Each of the methods rely on the estimation of an unknown parameter which is often estimated by maximum likelihood (although Bayesian versions are often used) requiring the Jacobian of the stabilization transform. The likelihood approach and development of the theoretical Jacobians and an empirical

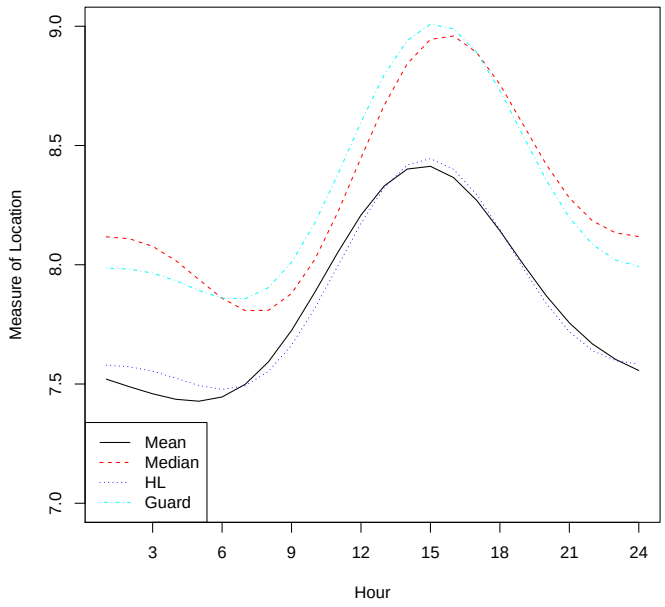

Figure 6: Functional measures of location for the Aberporth wind series.

comparison are presented in Section 10.

9 Transformations

9.1 Box-Cox Transform

Let $\mathbf{X}$ be an independent and identically distributed data set of size n, where $X_i \geq 0$, $\mathbb{E}X_i = \mu < \infty$ and $\operatorname{var} X_i = \sigma^2 < \infty$ for $i = 1, \ldots, n$. Box and Cox (1964) introduced the following parametric transform of the data:

$$Y_{\mathrm{BC},i}^{(\lambda)} = \begin{cases} (X_i^{\lambda} - 1)/\lambda & \text{for } \lambda \neq 0 \\ \log X_i & \text{for } \lambda = 0. \end{cases} \tag{9}$$

9.2 The Haar-Fisz Transform

The Haar-Fisz variance stabilizing transform bolts the Fisz transform, which pulls a pair of random variables towards the Gaussian, onto the discrete Haar wavelet transform. We review these two components next.

9.2.1 The Discrete Haar Wavelet Transform

The multiscale variance stabilization techniques defined in the next two sections are obtained by modifying the Haar wavelet transform. For further details see Vidakovic (1999) or Nason (2008) or many of the general books on wavelets such as Burrus et al. (1997), Daubechies (1992) or Mallat (1998).

Given data $X_1, \ldots, X_n$ where $n = 2^J$ for some integer $J > 1$ we define the Haar wavelet transform as follows. Set the initial coefficients $c_{J,i-1} = X_i$ for $i = 1, \ldots, n$. Then perform the recursive operation:

$$c_{j-1,i} = c_{j,2i} + c_{j,2i+1}, \tag{10}$$

and

$$d_{j-1,i} = c_{j,2i} - c_{j,2i+1}, \tag{11}$$

for $j = J, \ldots, 1$ and $i = 0, \ldots, 2^j - 1$. One often writes the coefficients at a given scale level, j, as a vector. Hence, $\mathbf{d}_j = (d_{j,0}, \ldots, d_{j,2^j-1})$ and $\mathbf{c}_j = (c_{j,0}, \ldots, c_{j,2^j-1})$. The full discrete Haar wavelet transform of $X_1, \ldots, X_n$ is the collection $\mathbf{d} = (\mathbf{c}_0, \mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{J-1})$. It is convenient to use the notation $\mathcal{H}$ to denote the Haar wavelet transform, so $\mathbf{d} = \mathcal{H}(\mathbf{X})$ where $\mathbf{X} = (X_1, \ldots, X_n)$.

The $\{d_{j,i}\}$ are known as mother wavelet coefficients and the $\{c_{j,i}\}$ are known as the father wavelet or scaling coefficients. Note that we have chosen a particular normalization for the Haar wavelet transform here. Definition 1 above uses $2^{-1}(1, -1)$ other expositions use $2^{-1/2}(1, -1)$, for example, as coefficients of the filtering operation in (11).

The transform can easily be inverted, the original data can be recovered only from the $\mathbf{d}$. The recursive steps (10) and (11) are simply reversed. In other words:

$$c_{j,2i} = (c_{j-1,i} + d_{j-1,i})/2, \tag{12}$$

and

$$c_{j,2i+1} = (c_{j-1,i} - d_{j-1,i})/2, \tag{13}$$

again for $j = 1, \ldots, J$ and $i = 0, \ldots, 2^j - 1$.

9.2.2 Haar-Fisz Transform

The Haar-Fisz transform was introduced in Fryzlewicz (2003), Fryzlewicz and Nason (2004) as a method for variance stabilization of data $P_1, \ldots, P_n$ distributed according to a Poisson distribution $P_i \sim \text{Poiss}(\lambda_i)$ for some sequence of intensities λ_i. Here λ_i was taken to be a sequence of samples from some function $\lambda(x)$ with given smoothness properties. The Haar-Fisz method was then adapted to χ^2-like data for local spectral estimation in Fryzlewicz and Nason (2006) and for spectral estimation for stationary time series in Fryzlewicz et al. (2008). The Haar-Fisz method was extended in a different direction by Fryzlewicz and Delouille (2005) and Fryzlewicz et al. (2007) by creating the 'data-driven Haar-Fisz' transformation which addressed problems where the mean-variance function, $h(\mu)$, is not known and has to be estimated from the data. For example, for Poisson data $h(\mu) = \mu$ and for χ^2-like $h(\mu) \propto \mu^2$. In an example, Fryzlewicz et al. (2007) estimated a two linear piece h for GOES satellite X-ray flux data's mean-variance function and postulated two mean-variance regimes linked to the autoranging electronics within the sensor. Further empirical evidence concerning the effectiveness of data-driven Haar-Fisz was demonstrated in Motakis et al. (2006) for variance stabilization of microarray data which also handled replicates and Nason and Bailey (2008) on estimation of conflict intensity. A detailed analysis of data-driven Haar-Fisz appears in Fryzlewicz (2008), theoretical work demonstrating asymptotic

normality for the inhomogeneous Poisson case appears in Schmidt and Xu (2008), and a generalization concerning wavelets/filters other than Haar was achieved by Jansen (2006).

The original Haar-Fisz method, for Poisson data, uses the following result by Fisz (1955):

Theorem 4 (Fisz) *Let* $P_i \sim \text{Poiss}(\lambda_i)$ *for* $i = 1, 2$ *and* X_1, X_2 *independent. Define the function* $\xi : \mathbb{R}^2 \to \mathbb{R}$ *by*

$$\xi(X_1, X_2) = \begin{cases} 0 & \text{if } X_1 = X_2 = 0, \\ (X_1 - X_2)/(X_1 + X_2)^{1/2} & \text{else.} \end{cases} \tag{14}$$

If $(\lambda_1, \lambda_2) \to (\infty, \infty)$ *and* $\lambda_1/\lambda_2 \to 1$ *then* $\xi(X_1, X_2) - \xi(\lambda_1, \lambda_2) \xrightarrow{D} N(0, 1)$.

The theorem shows that, under the right conditions, the quantity $\xi(X_1, X_2)$ is approximately Gaussian with a constant variance. The reader will note that $X_1 - X_2$ and $X_1 + X_2$ are merely the Haar mother and father wavelet coefficients of X_1, X_2. The innovation of the Haar-Fisz transform was to realize that one could replace the Haar mother coefficient in the Haar wavelet transform of section 9.2.1 by $\xi(X_1, X_2)$ and to do this recursively throughout the whole transform. Hence, one now possesses a wavelet tableaux where *all* the mother coefficients are now approximately Gaussian. This new tableaux is just a Haar wavelet transform containing these new coefficients which can be inverted. Since the transform is orthogonal the inverted transform will be approximately Gaussian with approximately constant variance. This is the Haar-Fisz transform, for Poisson data, and is fully invertible, just by reversing the steps.

To summarise, given a set of data $\mathbf{X}$ the steps in the Haar-Fisz transform for Poisson data are:

1. Apply the Haar wavelet transform to the data: $\mathbf{d} = \mathcal{H}(\mathbf{X})$.
2. Replace the mother wavelet Haar coefficients, $d_{j,k}$ by the Fisz-transformed equivalents $f_{j,k} = d_{j,k}/c_{j,k}^{1/2}$ to form $= (\mathbf{c_0}, \mathbf{f_0}, \ldots, \mathbf{f_{J-1}})$.
3. Invert the new wavelet coefficients to obtain the final transformed sequence $\mathbf{Y} = \mathcal{H}^{-1}(\mathbf{f})$.

Our article assumes that the data $X_1, \ldots, X_n$ are iid, but with no particular underlying parametric distribution in mind. The Haar-Fisz transform we introduce here lies somewhere between the fixed parametric assumptions in Fryzlewicz and Nason (2004, 2006) (Poisson and χ^2) and the more general data-driven mean-variance relationships found in Fryzlewicz and Delouille (2005); Fryzlewicz et al. (2007); Motakis et al. (2006) and Fryzlewicz (2008). Here, the assumption, as far as Haar-Fisz is concerned is $h(\mu) \propto \mu^{2\lambda}$. The appropriate value of λ for Poisson, then, is $\lambda = 1/2$ and for χ^2 we would have $\lambda = 1$. Our version of Haar-Fisz is conceptually similar to the single-parameter Box-Cox transform, but obviously Haar-Fisz is multiscale.

9.2.3 General form of our Haar-Fisz transform

We modify the general formula for the Haar-Fisz transform for Poisson data that appears in (Fryzlewicz, 2003, page 164) by adding a more general power transformation parameter λ as follows.

Let $\mathbf{X} = (X_1, \ldots, X_n)$ for $n = 2^J$ be the vector of interest. Introduce the family of Haar wavelet vectors $\{\psi^{j,k}\}$, where $j = 0, 1, \ldots, J-1$ is the scale parameter ($J-1$ is fine scale, 0 is coarsest) and $k = l2^{J-j}$, $l = 0, 1, \ldots, j$ is the location parameter. The components of of $\psi^{j,k}$ will be denoted by $\psi_i^{j,k}$ for $i = 0, \ldots, n-1$. We define:

$$\psi_i^{j,k} = \begin{cases} 0 & \text{for } i < k, \\ 1 & \text{for } k \leq i < k + 2^{J-j-1}, \\ -1 & \text{for } k + 2^{J-j-1} \leq i < k + 2^{J-j}, \\ 0 & \text{for } k + 2^{J-j} \leq i. \end{cases} \tag{15}$$

Similarly, we introduce the family of Haar scaling vectors $\{\phi^{j,k}\}$, whose components will be denoted by $\phi_i^{j,k}$ (the range of j, k and i remains unchanged). We define

$$\phi_i^{j,k} = \begin{cases} 0 & \text{for } i < k, \\ 1 & \text{for } k \leq i < k + 2^{J-j}, \\ 0 & \text{for } k + 2^{J-j} \leq i. \end{cases} \tag{16}$$

This definition of discrete Haar wavelets is similar to that of Nason et al. (2000). The difference is that we "pad" the wavelet vectors with zeroes on both sides so that they all have length n, and do not normalise them.

Further, let $\langle \cdot, \cdot \rangle$ denote the inner product of two vectors, and let $\mathbf{b}^J(i) = (b_0^J(i), b_1^J(i), \ldots, b_{J-1}^J(i))$ be the binary representation of the integer i, where $i < 2^J$.

The formula for the ith element of the Haar-Fisz transformed vector of $\mathbf{X}$, with parameter λ, is

$$U_i = \frac{< \phi^{0,0}, \mathbf{X} >}{n} + \sum_{j=0}^{J-1} (-1)^{b_j^J(i)} 2^{\frac{j-J}{2}} c_{j,J,i}(\mathbf{X}), \tag{17}$$

where

$$c_{j,J,i}(\mathbf{X}, \lambda) = \begin{cases} \frac{\langle \psi^{j, \lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X} \rangle}{\langle \phi^{j, \lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X} \rangle^\lambda} & \text{if } \langle \phi^{j, \lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X} \rangle > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

The only difference between (18) and Fryzlewicz (2003) is that we use a general λ whereas a fixed $\lambda = 1/2$ was used in Fryzlewicz (2003) to be used for Poisson distributed data. This is akin to the difference between the Anscombe (1948) transform and the Box-Cox transform with parameter λ.

We now adapt the general formula (18) into a slightly different form. Our adaption is useful for two purposes: (i) the new form conveniently encapsulates the next variance stabilization technique, described in section 9.3, (ii) the new form facilitates the establishment of an interesting new result concerning the Jacobians of both transforms.

Define the function $F_{\text{HF}} : [0, \infty)^3 \to \mathbb{R}$ by

$$F_{\text{HF}}(a, b, \lambda) = a(a + b)^{-\lambda}. \tag{19}$$

Then all of the terms in the general formula sum in (17) can be represented by the difference of two F_{HF} terms computed on the first and second half of the partial sum involved in that term. An example, for the case $n = 4$, should make this clear.

Suppose the data set is (X_1, X_2, X_3, X_4). Then let

$$\begin{aligned} Y^{(\lambda)}_{\text{HF},1} &= \tfrac{1}{4}\sum_{i=1}^{4} X_i + \frac{X_1 + X_2 - X_3 - X_4}{4(X_1 + X_2 + X_3 + X_4)^{\lambda}} + \frac{X_1 - X_2}{2(X_1 + X_2)^{\lambda}} \qquad (20) \\ &= \bar{X} + \tfrac{1}{4}F_{\text{HF}}(X_1 + X_2, X_3 + X_4, \lambda) - \tfrac{1}{4}F_{\text{HF}}(X_3 + X_4, X_1 + X_2, \lambda) \\ &+ \tfrac{1}{2}F_{\text{HF}}(X_1, X_2, \lambda) - \tfrac{1}{2}F_{\text{HF}}(X_2, X_1, \lambda). \qquad (21) \end{aligned}$$

The other three components of the Haar-Fisz transform are

$$Y^{(\lambda)}_{\text{HF},2} = \bar{X} + \frac{X_1 + X_2 - X_3 - X_4}{4(X_1 + X_2 + X_3 + X_4)^{\lambda}} - \frac{X_1 - X_2}{2(X_1 + X_2)^{\lambda}} \qquad (22)$$

$$Y^{(\lambda)}_{\text{HF},3} = \bar{X} - \frac{X_1 + X_2 - X_3 - X_4}{4(X_1 + X_2 + X_3 + X_4)^{\lambda}} + \frac{X_3 - X_4}{2(X_3 + X_4)^{\lambda}} \qquad (23)$$

$$Y^{(\lambda)}_{\text{HF},4} = \bar{X} - \frac{X_1 + X_2 - X_3 - X_4}{4(X_1 + X_2 + X_3 + X_4)^{\lambda}} - \frac{X_3 - X_4}{2(X_3 + X_4)^{\lambda}}, \qquad (24)$$

all of which can be put into a form similar to (21).

9.3 Multiscale Box-Cox Transform

More recently, Zhang et al. (2008) introduced a simple and elegant new variance stabilization technique that combines the discrete Haar wavelet transform with the well known Anscombe (1948) transform. Donoho (1993) first proposed denoising Poisson-distributed signals using wavelets by first applying Anscombe's transform, which results in approximately variance stabilized Gaussian data, and then using regular wavelet shrinkage for Gaussian data. Anscombe's transform, given by $\mathcal{A}(X_i) = \sqrt{X_i + 3/8}$, is essentially equivalent to preprocessing one's data with the Box-Cox transform with $\lambda = 1/2$

Zhang et al. (2008) begin by forming the Haar *father* wavelet coefficients by recursively applying formula (10) as normal. Then Anscombe's transform is applied to all of the father coefficients. Then, those Anscombe-transformed coefficients are used to form Haar *mother* wavelet coefficients. In other words, formula (11) becomes

$$d_i^{j-1} = \mathcal{A}(c_{2i-1}^{j}) - \mathcal{A}(c_{2i}^{j}). \qquad (25)$$

The beauty of their idea is that if the X_i are iid Poisson distributed then clearly so are all the c_k^j (as they are merely sums of independent Poissons). Hence, all the $\mathcal{A}(c_k^j)$ are approximately Gaussian with the approximately the same variance. Inversion of the new d_k^j Haar wavelet tableaux results in an approximately variance-stabilized Gaussian sequence. Both the Haar-Fisz transform and the transform introduced by Zhang et al. (2008) are similar in that they both produce stabilized Gaussian Haar wavelet coefficients which can then be inverted to provide variance stabilized data.

We generalize Zhang et al. (2008) by replacing Anscombe in (25) by the Box-Cox transform resulting in the *multiscale Box-Cox transform*. Define the function $F_{BC} : [0, \infty) \times \mathbb{R} \to \mathbb{R}$ by

$$F_{BC}(a, \lambda) = \begin{cases} (a^{\lambda} - 1)/\lambda & \text{for } \lambda \neq 0, \\ \log a & \text{for } \lambda = 0. \end{cases} \qquad (26)$$

For example, again for the case $n = 4$, we obtain:

$$\begin{aligned} Y^{(\lambda)}_{\text{MB},1} &= \bar{X} + \tfrac{1}{4}F_{BC}(X_1 + X_2, \lambda) - \tfrac{1}{4}F_{BC}(X_3 + X_4, \lambda) && (27) \\ &+ \tfrac{1}{2}F_{BC}(X_1, \lambda) - \tfrac{1}{2}F_{BC}(X_2, \lambda), && (28) \end{aligned}$$

similar formulae apply for $Y^{(\lambda)}_{\text{MB},2}$, $Y^{(\lambda)}_{\text{MB},3}$ and $Y^{(\lambda)}_{\text{MB},4}$. The full formula for $n = 8$ is presented as (54) in Appendix G. A general formula for the multiscale Box-Cox transform can be obtained by replacing $c_{j,J,n}$ in (18) by

$$\begin{aligned} f_{j,J,i}(\mathbf{X}, \lambda) &= F\{(\langle \psi^{j,\lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X}\rangle + \langle \phi^{J,\lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X}\rangle)/2, \lambda\} \\ &- F\{(\langle \psi^{j,\lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X}\rangle - \langle \phi^{J,\lfloor i/2^{J-j} \rfloor 2^{J-j}}, \mathbf{X}\rangle)/2, \lambda\}, \\ &= F\{(d_{j,J,i} + c_{j,J,i})/2, \lambda\} - F\{(d_{j,J,i} - c_{j,J,i})/2, \lambda\}, \qquad (29) \end{aligned}$$

where $c_{j,J,i}$ is as in (18), $d_{j,J,i}$ is the associated mother coefficient, $i = 0, \ldots, n-1$ and $F = F_{\text{BC}}$.

10 Likelihood

Section 9 defined the three transformations that we are considering in this paper: the Box-Cox, the Haar-Fisz and the multiscale Box-Cox. In practice, for any given set of data all of the transforms require the parameter λ to be chosen in some way. The seminal paper of Box and Cox (1964) introduced and studied a range of approaches to parameter estimation, including maximum likelihood and Bayesian methods, and these have become the standard and widely used methods across many fields. For brevity and focus we concentrate on analysis of the maximum likelihood approach here. However, a Bayesian analysis would be perfectly possible and desirable in many contexts.

The likelihood approach is explained in Atkinson (1985) whose clear approach we follow here. The aim of the likelihood approach is to choose λ that maximizes the Gaussian likelihood of the transformed observations but expressed as a function of the *original observations*, i.e.

$$(2\pi\sigma^2)^{-n/2} \exp\{-(Y^{(\lambda)} - X)^T (Y^{(\lambda)} - X)/2\sigma^2\} J, \qquad (30)$$

where $Y^{(\lambda)}, X$ are the vectorized versions of $Y_i^{(\lambda)}$ and X_i and where J is the Jacobian of the transformation, i.e.

$$J = \prod_{i=1}^{n} \left| \frac{\partial Y_i^{(\lambda)}}{\partial X_i} \right|. \qquad (31)$$

It is important to write the likelihood in terms of the original observations, which necessitates the use of the Jacobian to enable likelihoods from the same transformation, but with different λ, to be compared, and also to compare likelihoods from different transformations.

When the observations, X_i, are considered to be part of some model, e.g. $\mathbb{E}X = W\beta$, then the transformation approach needs to estimate both the transformation parameter, λ, the parameters of interest in the model, β and maybe σ^2, which has to be

estimated but might not be of direct interest. As described by Atkinson (1985) this can be achieved by a two-stage approach where the parameters (σ^2, β) are estimated in the normal way conditioned on λ to obtain the profile log-likelihood:

$$L_{\max}(\lambda) = -(n/2)\log\hat{\sigma}^2(\lambda) + \log J, \tag{32}$$

where $\hat{\sigma}^2 = n^{-1}Y^{(\lambda),T}(I-H)Y^{(\lambda)}$ is the usual maximum likelihood estimate of σ^2 and H is the usual hat matrix. Conveniently, this single approach will work with all of our transformations above. However, the exact likelihood and the form of Jacobian is different in each case. We shall address these details next. However, before we do it is worth nothing that the maximum likelihood framework provides important, useful and interesting information. For example, the asymptotic distribution of the maximizing parameter, confidence intervals for the parameter and convergence results.

10.1 Box-Cox likelihood

The Jacobian for the Box-Cox transform has a simple form due to the simplicity of the functional form of F_{BC} and the important fact that the Box-Cox transform is diagonal. In other words, the transformed value $Y_i^{(\lambda)}$ only depends on X_i and none of the other X_j for $j \neq i$. So,

$$\partial Y_i^{(\lambda)}/\partial X_i = \partial F_{\text{BC}}(X_i,\lambda)/\partial X_i = X_i^{\lambda-1}. \tag{33}$$

Hence, the (log) Jacobian is

$$\log J_{\text{BC}} = (\lambda-1)\sum_{i=1}^{n}\log X_i. \tag{34}$$

Observe that the log of the Jacobian is essentially the log of the geometric mean of the data, X: the multiscale measures of location arise from a similar observation made for the multiscale variance stabilizers below. Combining (32) with (34) one can see that the problem is one of penalized likelihood: the aim is to reduce the sample variance $\hat{\sigma}^2$ tensioned against an increasing (or decreasing if $\lambda < 1$) geometric mean of the data.

It seems that the penalized likelihood interpretation for the basic problem has not been emphasized in quite this way before, although it has appeared in more complex situations: such as using Box-Cox transformed curves in the estimation of reference centile curves in Cole and Green (1992). The penalized likelihood interpretation becomes increasingly useful and interesting when one considers the Jacobians of the Haar-Fisz and Multiscale Box-Cox transforms below.

10.2 Likelihood and Jacobian for multiscale transforms

A likelihood approach for multiscale variance transformation (Haar-Fisz) was used by Nason and Bailey (2008) although for correlated time series data. The associate editor on that publication inspired the current article by remarking that it would be fascinating if a multiscale variance transform could be shown to dominate Box-Cox. Section 10.5 shows that multiscale does *not* dominate Box-Cox, but they do almost always dominate in terms of likelihood (stabilization) although not necessarily in terms of normalization (but only in terms of the limited simulation study presented there).

For the theory the Jacobians for the Haar-Fisz and Multiscale Box-Cox transforms (="the multiscale transforms") are considerably more difficult to establish because the transforms are not diagonal. This section establishes a general result for the multiscale transform Jacobians which makes use of the fact that both Haar-Fisz and Multiscale Box-Cox arise from the same general form given in (17). They both show that the likelihood maximization is essentially a penalized optimization with the penalty in all cases related to a measure of location (which inspired the first part of this article).

Definition 6 *Define the general multiscale variance stabilization transform $Y^{(\lambda)}$ of X by the general sum given in* (17) *with the new $f_{j,J,n}$ given in* (29) *with $F = F_{HF}$ for Haar-Fisz and $F = F_{BC}$ for the Multiscale Box-Cox transform.*

Theorem 5 *Define $\mathcal{T}$ to be the set of all unique possible terms in the general sum in* (17) *for all $i = 0, \ldots, n-1$. The Jacobian of the multiscale variance stabilization transform given in Definition 6 is given by*

$$J(F, \lambda) = 2^{n-1} \prod_{j \in \mathcal{T}} F\{(d_{j,J,n} + c_{j,J,n})/2, \lambda\} + F\{(d_{j,J,n} - c_{j,J,n})/2, \lambda\}. \quad (35)$$

The proof of Theorem 5 appears in Appendix F. The proof is constructive and makes heavy use of the dyadic structure in the Jacobian which arises because of the binary/dyadic construction of the general formula in (17). An example of the Jacobian for the Multiscale Box-Cox transform is given in Appendix G.

All of our theoretical Jacobians below have been checked for correctness by comparing to the result of a numerical Jacobian procedure adapted from the `fdjac` routine from Press et al. (1992, Page 388).

10.3 Haar-Fisz Jacobian and the Haar-Fisz geometric mean

Let $c_{j,k}$ be the father Haar wavelet coefficients defined in (10). Then the general Jacobian (35) for the Haar-Fisz transform on choosing $F = F_{\text{HF}}$ further simplifies to

$$J_{\text{HF}}(\lambda) = J(F_{\text{HF}}, \lambda) = 2^{n-1} \prod_{j=0}^{J-1} \prod_{k=0}^{2^j-1} c_{j,k}^{-\lambda}. \quad (36)$$

Remarkably, the Jacobian of the rather complicated non-diagonal Haar-Fisz transform is merely the the product of the father wavelet coefficients raised to the power of $-\lambda$.

Just as the Jacobian for the regular Box-Cox transform (for $\lambda = 1$) is proportional to the geometric mean of the data, the quantity $J(F_{\text{HF}}, 1)$ is proportional to the (reciprocal of the) geometric mean of the father wavelet coefficients of the data. This is the origin of the umm_{GM} estimator introduced in Example 4.

10.4 Multiscale Box-Cox Jacobian and its geometric mean

The Multiscale Box-Cox transform's Jacobian also simplifies on choosing $F = F_{\text{BC}}$ in (35) to give

$$J(F_{\text{BC}}, \lambda) = 2^{n-1} \prod_{j=1}^{J} \prod_{k=0}^{2^{j-1}-1} (c_{j,2k}^{\lambda-1} + c_{j,2k+1}^{\lambda-1}). \quad (37)$$

Table 2: Number of times particular stabilization method achieves maximum likelihood out of 100 trials.

Distribution	Box-Cox	Haar-Fisz	Multiscale BC
Poisson	0	67	33
Log-Normal	1	6	93
Folded Normal	0	11	89
χ^2	0	0	100
Geometric	0	6	94

For example, if $n = 4$ then

$$J(F_{\text{BC}}, \lambda) = 8\{(X_1 + X_2)^{\lambda-1} + (X_3 + X_4)^{\lambda-1}\}(X_1^{\lambda-1} + X_2^{\lambda-1})(X_3^{\lambda-1} + X_4^{\lambda-1}). \tag{38}$$

As in the previous section, for the special value of $\lambda = 2$, since the $c_{j,k}$ are arithmetic means of parts of the data set at different scales and locations, the sum of $c_{j,2k} + c_{j,2k+1}$ in (37) is proportional to yet another arithmetic mean, and the Jacobian $J(F_{\text{BC}}, 2)$ is related to a geometric mean of all of those. Indeed, setting $\lambda = 2$ and using (10) we obtain

$$J(F_{\text{BC}}, 2) = 2^{n-1} \prod_{j=1}^{J} \prod_{k=0}^{2^{j-1}-1} c_{j-1,k} = 2^{n-1} \prod_{j=0}^{J-1} \prod_{k=0}^{2^j-1} c_{j,k}. \tag{39}$$

Hence, the associated measure of location for the multiscale Box-Cox transform is the same as for the Haar-Fisz transform.

10.5 Comparison of Stabilization Methods

With current mathematical techniques the only way to maximize the stabilization likelihood in (32) is via numerical methods, this is true for the Box-Cox transform as well as the two multiscale transforms introduced above. This section provides a numerical comparison on how good the transforms are in two ways: by comparing their optimized likelihood values and also how 'Gaussian' the transformed samples are (via the Shapiro-Wilk normality test, `shapiro.test` in R) on five different distributions. Each run draws $n = 64$ iid random variables from the distribution the Box-Cox, Haar-Fisz and multiscale Box-Cox transforms are applied and the stabilization technique with the largest likelihood and most Gaussian Shapiro-Wilk result is noted. Table 2 shows the result of maximising the likelihood (32) for samples of size 64 from five different distributions, each for 100 runs. The multiscale Box-Cox transform performs the best for all distributions apart from the Poisson, where the Haar-Fisz transform works best.

Table 3, the 'most Gaussian' according to Shapiro-Wilk, shows a more mixed picture. Haar-Fisz dominates for the Poisson and Geometric distributions and regular Box-Cox 'wins' for Log-Normal, Folded Normal and χ^2. The distributions were: (a) $X_i \sim$ Poisson$(3) + 1$; (b) $X_i \sim \exp(Z_i)$, where $Z_i \sim N(1, 1)$, lognormal; (c) $X_i \sim |Z_i|$, where $Z_i \sim N(1, 1)$, folded normal; (d) χ^2_1; (e) $X_i \sim$ Geometric$(0.2) + 1$.

Table 3: Number of times particular stabilization method achieves most Gaussian Shapiro-Wilk test p-value out of 100 trials.

Distribution	Box-Cox	Haar-Fisz	Multiscale BC
Poisson	2	65	33
Log-Normal	55	36	9
Folded Normal	61	7	32
χ^2	37	31	32
Geometric	0	98	2

Acknowledgements

This work was partially supported by EPSRC grant EP/D005221/1 and The Energy Programme. The Energy Programme is an RCUK cross-council initiative led by EPSRC and contributed to by ESRC, NERC, BBSRC and STFC. I am grateful to Kerry Bristow for pointing out the possibility of triadic multimeans. All the computations in this paper were carried out in R, R Development Core Team (2009).

A Proof of Theorem 1

i. We first prove $\mathrm{GM}(\mathbf{X}) \leq \mathrm{umm}_{\mathrm{GM}}(\mathbf{X})$. For $n = 1$ the result is trivial. The result relies on the concavity of the $\log$ function and equation (2) from Definition 1 to show that:

$$\log c_{j-1,i} \geq \tfrac{1}{2} \log c_{j,2i} + \tfrac{1}{2} \log c_{j,2i+1}, \tag{40}$$

with equality iff $c_{j,2i} = c_{j,2i+1}$. For $n > 2$ we have

$$\begin{aligned}
\log \mathrm{umm}_{\mathrm{GM}}(\mathbf{X}) &= (n-1)^{-1} \sum_{j=0}^{J-1} \sum_{k=0}^{2^j-1} \log c_{j,k} \\
&= (n-1)^{-1} \left(\sum_{j=1}^{J-1} \sum_{k=0}^{2^j-1} \log c_{j,k} + \log c_{0,0} \right) \\
&\geq (n-1)^{-1} \left(\sum_{j=1}^{J-1} \sum_{k=0}^{2^j-1} \log c_{j,k} + \tfrac{1}{2} \log c_{1,0} + \tfrac{1}{2} \log c_{1,1} \right) \\
&= (n-1)^{-1} \left(\sum_{j=2}^{J-1} \sum_{k=0}^{2^j-1} \log c_{j,k} + \tfrac{3}{2} \log c_{1,0} + \tfrac{3}{2} \log c_{1,1} \right) \\
&\geq (n-1)^{-1} \left(\sum_{j=2}^{J-1} \sum_{k=0}^{2^j-1} \log c_{j,k} + \tfrac{3}{4} \sum_{r=0}^{3} \log c_{2,r} \right) \\
&= (n-1)^{-1} \left(\sum_{j=3}^{J-1} \sum_{k=0}^{2^j-1} \log c_{j,k} + \tfrac{7}{4} \sum_{r=0}^{3} \log c_{2,r} \right) \\
&\geq (n-1)^{-1} \left(\sum_{j=4}^{J-1} \sum_{k=0}^{2^j-1} \log c_{j,k} + \tfrac{15}{8} \sum_{r=0}^{7} \log c_{3,r} \right). \qquad (41)
\end{aligned}$$

Then, continuing in this way, successively replacing coarser $c_{j,k}$ with finer ones, we end up with:

$$\begin{aligned}
\log \mathrm{umm}_{\mathrm{GM}}(\mathbf{X}) &\geq (n-1)^{-1} \frac{2^J - 1}{2^J} \sum_{k=0}^{2^J-1} \log c_{J,k} \qquad (42) \\
&= n^{-1} \sum_{k=1}^{n} \log x_k = \log \mathrm{GM}(\mathbf{X}), \qquad (43)
\end{aligned}$$

with equality iff all the entries in $\mathbf{X}$ are identical. For the transition from (42) to (43) recall that $c_{J,k} = X_{k+1}$ for $k = 0, \ldots, n-1$ mentioned immediately before (2).

ii. Now we prove $\mathrm{umm}_{\mathrm{GM}}(\mathbf{X}) \leq \bar{X}$. For $n = 1$ the result is trivial. For $n = 2$ it is easy to see that $\bar{X} = \mathrm{umm}_{\mathrm{GM}}(\mathbf{X}) = (X_1 + X_2)/2$. For $n > 2$ (but still $n = 2^J$) we first establish the subsidiary result $(n-2) \log c_{0,0} \geq \sum_{j=1}^{J-1} \sum_{k=0}^{2^j-1} \log c_{j,k}$ by again

using the concavity result (40)

$$\begin{aligned}
(n-2)\log c_{0,0} &\geq \left(\tfrac{n-2}{2}\right)\sum_{r=0}^{1}\log c_{1,r} = \sum_{r=0}^{1}\log c_{1,r} + \left(\tfrac{n-4}{2}\right)\sum_{r=0}^{1}\log c_{1,r} \\
&\geq \sum_{r=0}^{1}\log c_{1,r} + \left(\tfrac{n-4}{4}\right)\sum_{r=0}^{3}\log c_{2,r} \\
&= \sum_{j=1}^{2}\sum_{r=0}^{2^j-1}\log c_{j,r} + \left(\tfrac{n-8}{4}\right)\sum_{r=0}^{3}\log c_{2,r} \\
&\geq \sum_{j=1}^{p}\sum_{r=0}^{2^j-1}\log c_{j,r} + \left(\frac{n-2^{p+1}}{2^p}\right)\sum_{r=0}^{2^p-1}\log c_{p,r} \qquad (44) \\
&\geq \sum_{j=1}^{J-1}\sum_{r=0}^{2^j-1}\log c_{j,r}. \qquad (45)
\end{aligned}$$

where p can range from 1 to $J-1$ in (44) and indeed $p = J-1$ in (45). To complete the result we first take exponentials of both sides of (45) to obtain $c_{0,0}^{n-2} \geq \prod_{j=1}^{J-1}\prod_{r=0}^{2^j-1} c_{j,r}$. Then take the $(n-1)$th root of both sides to obtain

$$c_{0,0}^{1-\frac{1}{n-1}} = c_{0,0}^{\frac{n-2}{n-1}} \geq \left(\prod_{j=1}^{J-1}\prod_{r=0}^{2^j-1} c_{j,r}\right)^{\frac{1}{n-1}}. \qquad (46)$$

and multiply both sides by $c_{0,0}^{\frac{1}{n-1}}$ to obtain

$$\bar{X} = c_{0,0} \geq \left(\prod_{j=0}^{J-1}\prod_{r=0}^{2^j-1} c_{j,r}\right)^{\frac{1}{n-1}} = \mathrm{umm}_{\mathrm{GM}}(\mathbf{X}), \qquad (47)$$

as required. Once more equality holds if all the entries in $\mathbf{X}$ are identical this is because equality holds in (40) in this circumstance.

B Proof of Proposition 1

We first examine the case $n = 4$ and set $x_1 = a, x_2 = b, x_3 = c$ and $x_4 = d$ and assume $a < b < c < d$. With $n = 4$ there are three Haar wavelet coefficients: $c_{1,0} = (a+b)/2, c_{1,1} = (c+d)/2$ and $c_{0,0} = (a+b+c+d)/4$. First, note that $c_{0,0}$ is invariant to any permutation of (a, b, c, d) to (x_1, x_2, x_3, x_4). Further, switching a with b, or switching c with d does not change $c_{1,0}$ or $c_{1,1}$ respectively. Since the multimean is a function of the product of the $c_{j,k}$ the only change in the multimean is produced by moving a (or b) from $c_{1,0}$ to $c_{1,1}$ and moving c (or d) in the other direction. Without loss of generality let us consider switching b with c and examine the product $c_{1,0}c_{1,1}$ in both cases (since $c_{0,0}$ would be unaltered). In the sorted case (a, b contributes to $c_{1,0}$ and c, d to $c_{1,1}$) we have the product:

$$\text{sorted} = c_{1,0}c_{1,1} = \tfrac{1}{4}(a+b)(c+d). \qquad (48)$$

In the switched case we have:

$$\text{switched} = c_{1,0}c_{1,1} = \tfrac{1}{4}(a+c)(b+d). \tag{49}$$

We show that the 'sorted' is smaller than the 'switched':

$$\text{sorted-switched} = ac + bd - ab - cd = (a-d)(c-b) < 0, \tag{50}$$

since $a - d < 0$ and $c - b > 0$. Note that the result would be the same if we had interchanged either a, b with either of c, d. For general $n = 2^J$ this principle for $n = 4$ can be extended to all levels of Haar wavelet coefficients as the calculation of these coefficients in (2) is recursive and identical at every step. In other words, if at any level j the 'data' (or Haar father coefficients at that level) are out of order then the coefficients computed at level $j-1$ can always be made smaller by reordering the data coefficients into ascending order which can be achieved by ordering the input data into ascending order.

C Proof of Theorem 2

The estimator $\mathrm{mm}_{\mathrm{median}}$ is the median of the set of multiscale means, $\mathcal{C}(\mathbf{X}_{p^*})$, of the order statistics $X_{(1)}, \ldots, X_{(n)}$. Recall that n is even and for this proof write $\mathcal{C}$ for $\mathcal{C}(\bar{X}_{p^*})$ for brevity and note that the number of elements in $\mathcal{C}$ is $n-1$. Partition all elements of the set $\mathcal{C}$ into two subsets: $\mathcal{C}_1 = \{\bar{X}_{[s:t]} \in \mathcal{C} : t \leq n/2\}$ and $\mathcal{C}_2 = \{\bar{X}_{[s:t]} \in \mathcal{C} : s > n/2\}$. Then $\mathcal{C} = \{\bar{X}\} \cup \mathcal{C}_1 \cup \mathcal{C}_2$ and $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$. Note that $\bar{X}$ is not a member of $\mathcal{C}_1$ nor $\mathcal{C}_2$ since $\bar{X} = \bar{X}_{[1:n]} = c_{0,0}$.

We now show that $c \leq \bar{X}_{[(n/2-1):n/2]}$, the left guard value, for all $c \in \mathcal{C}_1$. We know that $c \in \mathcal{C}_1$ is a dyadic mean of, at most, 2^{j-1} consecutive values taken somewhere from $X_1, \ldots, X_{n/2}$. Suppose the number of observations involved in the mean, c, is 2^k where $k \leq j-1$. Then $c = 2^{-k}(X_{(i)} + \cdots + X_{(i+2^k-1)})$, where $i + 2^k - 1 \leq n/2$ as $c \in \mathcal{C}_1$. Then:

$$c \leq 2^{-k}(2^{k-1}X_{(n/2-1)} + 2^{k-1}X_{(n/2)}) = 2^{-1}(X_{(n/2-1)} + X_{(n/2)}) = \bar{X}_{[(n/2-1):n/2]}, \tag{51}$$

this is because $X_{(i)} \leq X_{(n/2-1)}$, and then for $X_{(i+1)}$ and so on, as they are order statistics and $c \in \mathcal{C}_1$. Similarly, we can show that $c \geq \bar{X}_{[(n/2+1):(n/2+2)]}$ for $c \in \mathcal{C}_2$.

Now denote the $(n-4)/2$ values in $\mathcal{C}_1$ apart from $\bar{X}_{[(n/2-1):n/2]}$ by c_i^1 for $i = 1, \ldots, (n-4)/2$ and those in $\mathcal{C}_2$ apart from $\bar{X}_{[(n/2+1):(n/2+2)]}$ by c_i^2 for $i = 1, \ldots, (n-4)/2$. Then we can order all $c \in \mathcal{C} \setminus \{\bar{X}\}$ by

$$c_{(1)}^1 \leq \cdots \leq c_{((n-4)/2)}^1 \leq \bar{X}_{[(n/2-1):n/2]} \leq \bar{X}_{[(n/2+1):(n/2+2)]} \leq c_{(1)}^2 \cdots c_{((n-4)/2)}^2. \tag{52}$$

To discover the median of the full set of $n-1$ coefficients in $\mathcal{C}$ we need to insert $\bar{X}$ somewhere into the inequality sequence in (52). It is clear that if $\bar{X} < \bar{X}_{[(n/2-1):n/2]}$ then the median of the whole new sequence (with $\bar{X}$ inserted) is $\bar{X}_{[(n/2-1):n/2]}$. If $\bar{X} > \bar{X}_{[(n/2+1):(n/2+2)]}$ then the median is $\bar{X}_{[(n/2+1):(n/2+2)]}$. Finally, if $\bar{X}_{[(n/2-1):n/2]} \leq \bar{X} \leq \bar{X}_{[(n/2+1):(n/2+2)]}$ then the median value is $\bar{X}$. Hence, result.

D Proof of Proposition 2

Without loss of generality to show expectation assume $\mu = 0$. Define the indicator variable I such that $I = 1$ when $\bar{X} < X_{(\ell)}$, $I = 2$ when $\bar{X} > X_{(r)}$, and $I = 3$ when $X_{(\ell)} \leq \bar{X} \leq X_{(r)}$, where $X_{(\ell)}, X_{(r)}$ are the left and right guard values respectively. Then

$$\begin{aligned}
\mathbb{E}\{\text{guard}(\mathbf{X})\} &= \mathbb{E}\{\text{median}(X_{(\ell)}, \bar{X}, X_{(r)})\} \\
&= \sum_{i=1}^{3} \mathbb{E}\{\text{guard}(\mathbf{X}) | I = i\}\mathbb{P}(I = i) \\
&= \mathbb{E}\{X_{(\ell)} | \bar{X} < X_{(\ell)}\}\mathbb{P}(I = 1) + \mathbb{E}\{X_{(r)} | \bar{X} > X_{(r)}\}\mathbb{P}(I = 2) \\
&+ \mathbb{E}\{\bar{X} | X_{(\ell)} < \bar{X} < X_{(r)}\}\mathbb{P}(I = 3).
\end{aligned}$$

By symmetry we have that $\mathbb{P}(I = 1) = \mathbb{P}(I = 2)$ and $\mathbb{E}\{X_{(\ell)} | \bar{X} < X_{(\ell)}\} = -\mathbb{E}\{X_{(r)} | \bar{X} > X_{(r)}\}$. Hence,

$$\mathbb{E}\{\text{guard}(\mathbf{X})\} = \mathbb{E}\{\bar{X} | X_{(\ell)} < \bar{X} < X_{(r)}\}\mathbb{P}(I = 3),$$

and by symmetry again $\mathbb{E}\{\bar{X} | X_{(\ell)} < \bar{X} < X_{(r)}\} = 0$. Hence, $\mathbb{E}\{\text{guard}(\mathbf{X})\} = 0$.

For the variance we note that:

$$\bar{X}_{[(n/2-1):n/2]} \leq \text{guard}(\mathbf{X}) \leq \bar{X}_{[(n/2+1):(n/2+2)]}, \tag{53}$$

for all n. Arnold et al. (1992, Theorem 8.5.1) shows that, for $0 < p < 1$, $X_{(\lceil np \rceil)}$ is asymptotically normal with mean $F^{-1}(p)$ and variance $p(1-p)/(n[f\{F^{-1}(p)\}]^2)$. Hence, the left guard value, $\bar{X}_{[(n/2-1):n/2]}$ is also asymptotically normal with mean $\{F^{-1}(1/2 - 1/n) + F^{-1}(1/2)\}/2$ which tends to μ as $n \to \infty$ due to symmetry. The same can be shown for the right guard value.

Now write (53) as $c_n \leq a_n \leq b_n$ and μ_a, μ_b and μ_c for the means of the random variables $a_n, b_n, c_n \in \mathbb{R}$ respectively. Then standard theory shows

$$\begin{aligned}
a_n^2 \leq \max(b_n^2, c_n^2) &\implies a_n^2 - \mu_a^2 \leq \max(b_n^2 - \mu_a^2, c_n^2 - \mu_a^2) \\
&\implies \text{var}(a_n) \leq \max\{\mathbb{E}(b_n - \mu_a^2), \mathbb{E}(c_n - \mu_a^2)\} \\
&\implies \text{var}(a_n) \leq \max\{\text{var}(b_n) + \mu_b^2 - \mu_a^2, \text{var}(c_n) + \mu_c^2 - \mu_a^2\} \\
&\implies \text{var}(a_n) \to 0,
\end{aligned}$$

because $\text{var}(b_n), \text{var}(c_n) \to 0$, $\mu_a = \mu$ and $\mu_b, \mu_c \to \mu$ due to the asymptotic normality result described by Arnold et al. (1992). Hence, $\text{guard}(\mathbf{X})$ is (mean square) consistent for μ.

E Proof of Theorem 3

Here $n = 2^J$ which means that there are an odd number, $n - 1$, of coefficients, $\mathcal{C}$, formed from order statistics from the data. The guard estimator computes the median of these $n - 1$ coefficients. Hence, breakdown occurs when at least $n/2$ of the coefficients get pulled to infinity.

Starting with the original data, to determine the breakdown point, we imagine the data points being pulled to infinity one-by-one. Since guard is computed from order statistics this is equivalent to assuming that the first data point to go to infinity is $X_{(n)}$, then $X_{(n-1)}$, and so on. Since the coefficients are ordered (as they are themselves formed from order statistics) breakdown first occurs when the finest scale coefficient corresponding to $\bar{X}_{[(n/2+1):(n/2+2)]}$ is pulled to infinity: at this point there are $n/2-1$ larger coefficients which have already been pulled to infinity ('larger' due to similar arguments used in the proof of Theorem 2 above). Hence, when $X_{(n/2+2)}$ is pulled to infinity breakdown occurs, i.e. when $X_{(n/2+2)}, X_{(n/2+3)}, \ldots, X_{(n)}$ have been pulled to infinity. Hence, the minimal number of observations that have to be 'pulled' to achieve breakdown is $n-(n/2+2)+1 = n/2-1$, which is $\frac{1}{2}-\frac{1}{n}$ expressed as a proportion of n.

F Proof of Theorem 5

Our proof begins by considering the final terms of the general sum for the multiscale transforms, essentially (17) with the $f_{j,J,n}$ coefficient given by (21) and (29) for Haar-Fisz and Multiscale Box-Cox respectively. Let us temporarily concentrate on the Multiscale Box-Cox transform. The last terms in the sum, for each successive i are $a_1 = +\{f_{\mathrm{BC}}(X_1) - f_{\mathrm{BC}}(X_2)\}$, $a_2 = -\{f_{\mathrm{BC}}(X_1) - f_{\mathrm{BC}}(X_2)\}$ (for $i = 1, 2$), then $a_3 = +\{f_{\mathrm{BC}}(X_3) - f_{\mathrm{BC}}(X_4)\}$, $a_4 = -\{f_{\mathrm{BC}}(X_3) - f_{\mathrm{BC}}(X_4)\}$ (for $i = 3, 4$), up to $a_{n-1} = +\{f_{\mathrm{BC}}(X_{n-1}) - f_{\mathrm{BC}}(X_n)\}$, $a_n = -\{f_{\mathrm{BC}}(X_{n-1}) - f_{\mathrm{BC}}(X_n)\}$ (for $i = n-1, n$). It is important to note that $a_1 = -a_2$, $a_3 = -a_4$, and so on.

To construct the Jacobian all of the above n terms are each to be differentiated by ∂X_j for $j = 1, \ldots, n$. The differentiation of a_1, a_2 by X_j is only non-zero for $j = 1, 2$, resulting in $a_{1,1}$ and $a_{2,2}$, and similarly for all the other terms. Hence, in the Jacobian the term involving the differential of a_1, a_2 only appears in two rows and as $a_1 = -a_2$ this term in that row is of opposite sign in the two rows. The same thing happens for all of the other pairs a_{2i-1}, a_{2i}, but the terms only appear at the places where their differential with respect to X_j is nonzero. Precisely the same pattern occurs with the Haar-Fisz transform, although each differential term is a fraction with a denominator containing the $X_1 + X_2$ sum.

A similar pattern emerges for the coarser scale coefficients, but with more repeated rows. For example, for the next most 'coarsest' terms in the general sum we have, for successive i: $b_1 = +\{f_{\mathrm{BC}}(\sum_1^2 X_k) - f_{\mathrm{BC}}(\sum_3^4 X_k)\}$, $b_2 = b_1$, $b_3 = -b_1$, $b_4 = -b_1$ and then $b_5 = +\{f_{\mathrm{BC}}(\sum_5^6 X_k) - f_{\mathrm{BC}}(\sum_7^8 X_k)\}$, $b_6 = b_5$, $b_7 = -b_5$, $b_8 = -b_5$ and so on up to $b_{n-3} = +\{f_{\mathrm{BC}}(\sum_{n-3}^{n-2} X_k) - f_{\mathrm{BC}}(\sum_{n-1}^{n} X_k)\}$, $b_{n-2} = b_{n-3}$, $b_{n-1} = -b_{n-3}$, $b_n = -b_{n-3}$.

These terms enter into the Jacobian and are each differentiated by ∂X_j for $j = 1, \ldots, n$. The differentiation of $b_1, \ldots, b_4$ by X_j are only non-zero for $j = 1, \ldots, 4$, and similarly for other terms. So, these terms, after differentiation occur in blocks of four rows each. The first two of the four are the same, and the second two are the negative of those).

Similar patterns emerge for the coefficients of coarser scales still (i.e. the next would occur in blocks of 8, with four rows of terms the same, and the next four the negative of those, and so on.). The first term in the general sum is the "sum of all the

coefficients" which when differentiated gives a constant which is added to every term in the Jacobian.

Hence, we end up with a Jacobian with a great deal of structure. The highly-organized structure suggests a pattern of row and column operations that can considerably simplify the Jacobian. Given the dyadic block nature of the structure described above, the pattern of operations follows a binary pattern. Let R_k denote the kth row, and C_k denote the kth column. The notation $x \to y$ means row (or column) x replaces row (or column) y. The operations are:

1. Perform $R_{2k} + R_{2k-1} \to R_{2k-1}$ for $k = 1, \ldots, 2^{J-1}$. This cancels out the finest scale terms in every odd row, i.e. the a_2 neutralizes the a_1 in the first row. This operation also doubles the values of all of the other terms in the odd rows (as the even row contains the same information at the coarser scales, it is just the finest scale information that differs on successive rows). Hence, a factor of 2 can be extracted from all the odd rows. Taken for all the rows a scale factor of 2^{J-1} can be extracted.

2. Then perform $R_{2k} - R_{2k-1} \to R_{2k}$ for $k = 1, \ldots, 2^{J-1}$. All of the information at scales coarser than the finest cancels out. Each of the even rows contains only two non-zero columns which contain $(-a_{1,1}\ a_{2,2})$, $(-a_{3,3}\ a_{4,4})$ and so on up until the last row which contains all zeroes followed by $(-a_{n-1}\ a_n)$.

3. Then perform $R_{4k-1} + R_{4k-3} \to R_{4k-3}$ for $k = 1, \ldots, 2^{J-2}$. This cancels out the next coarser scale information at rows R_{4k-3} similar to step 1 for the finer scales. Similarly, we can extract a factor of 2^{J-2} at this point, making the total extracted factor 2^{2J-3}. Then perform $R_{4k-1} - R_{4k-3} \to R_{4k-1}$ as in step 2. This results in 2^{J-2} rows at R_{4k-1} which are all zeroes apart from blocks of four coefficients corresponding to $(-b_{1,1}\ -b_{1,1}\ b_{2,2},\ b_{2,2})$ and so on (actually, the second $b_{1,1}$ is formally $b_{1,2}$ but this is equal to $b_{1,1}$, and so on).

4. Then perform $R_{8k-1} + R_{8k-7} \to R_{8k-7}$ for $k = 1, \ldots, 2^{J-3}$, this cancels out the next coarser scale and enables another factor of 2^{J-3} to be extracted resulting in a total extracted factor of 2^{3J-6}. Then perform $R_{8k-1} + R_{8k-7} \to R_{8k-1}$ as in the previous steps. This results in 2^{J-3} rows at R_{8k-1} which are all zeroes apart from blocks of eight coefficients.

5. These steps should be continued until row R_{n-1} is reached and processed. The extracted factor at this stage is $2^{J-1}2^{J-2}\cdots 2 = 2^{2^J-1} = 2^{n-1}$.

6. Steps 1. to 5. are now applied to the *columns* of the Jacobian, but because of the abundancy of zeroes no doubling of values of rows occur and now extra factors of two are extracted. This results in a Jacobian where (i) the top row consists of a single 1 followed by $n-1$ zeros; (ii) the diagonal of the Jacobian consists of the top-left 1 just mentioned and each of the terms in formula in (35): one for each scale and location apart from the very coarsest (hence n of them); (iii) a sparse arrangement of off-diagonal elements.

7. We now expand the determinant using Laplace's method by successively pivoting on the diagonal elements only (and that is why they end up in the product

in (35). The ordering is carried out from finer scale to coarser scale coefficients. Expand the determinant in the following order (i) pivot on (1,1), which has the single entry of 1 and a row of zeroes; (ii) then pivot successively on the elements which have the finest scale $f_{j,J,k}$ coefficients in the diagonal. These are all in columns with all other entries zero and there are 2^{J-1} of them, half of the columns. This will result in a Jacobian with all the $f_{j,J,n}$ coefficients at the next coarsest scale, and these too will now be in columns consisting of entirely of zeroes apart from the diagonal entry; (iii) pivot on the next coarsest scale elements, and so on.

G Example of Jacobian result

We follow the steps of the constructive proof presented in Appendix F for the case $n = 2^J = 8$, $J = 3$ for the Multiscale Box-Cox transform where the data is $X_1, \ldots, X_8$. The Multiscale Box-Cox transform for $i = 1, \ldots, n$ and parameter λ is

$$\begin{aligned}
Y_1 &= \Sigma X_i + \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&+ \{f_{\text{BC}}(\Sigma_1^2 X_i, \lambda) - f_{\text{BC}}(\Sigma_3^4 X_i, \lambda)\} + \{f_{\text{BC}}(X_1, \lambda) - f_{\text{BC}}(X_2, \lambda)\}, \\
Y_2 &= \Sigma X_i + \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&+ \{f_{\text{BC}}(\Sigma_1^2 X_i, \lambda) - f_{\text{BC}}(\Sigma_3^4 X_i, \lambda)\} - \{f_{\text{BC}}(X_1, \lambda) - f_{\text{BC}}(X_2, \lambda)\}, \\
Y_3 &= \Sigma X_i + \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&- \{f_{\text{BC}}(\Sigma_1^2 X_i, \lambda) - f_{\text{BC}}(\Sigma_3^4 X_i, \lambda)\} + \{f_{\text{BC}}(X_3, \lambda) - f_{\text{BC}}(X_4, \lambda)\}, \\
Y_4 &= \Sigma X_i + \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&- \{f_{\text{BC}}(\Sigma_1^2 X_i, \lambda) - f_{\text{BC}}(\Sigma_3^4 X_i, \lambda)\} - \{f_{\text{BC}}(X_3, \lambda) - f_{\text{BC}}(X_4, \lambda)\}, \\
Y_5 &= \Sigma X_i - \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&+ \{f_{\text{BC}}(\Sigma_5^6 X_i, \lambda) - f_{\text{BC}}(\Sigma_7^8 X_i, \lambda)\} + \{f_{\text{BC}}(X_5, \lambda) - f_{\text{BC}}(X_6, \lambda)\}, \\
Y_6 &= \Sigma X_i - \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&+ \{f_{\text{BC}}(\Sigma_5^6 X_i, \lambda) - f_{\text{BC}}(\Sigma_7^8 X_i, \lambda)\} - \{f_{\text{BC}}(X_5, \lambda) - f_{\text{BC}}(X_6, \lambda)\}, \\
Y_7 &= \Sigma X_i - \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&- \{f_{\text{BC}}(\Sigma_5^6 X_i, \lambda) - f_{\text{BC}}(\Sigma_7^8 X_i, \lambda)\} + \{f_{\text{BC}}(X_7, \lambda) - f_{\text{BC}}(X_8, \lambda)\}, \\
Y_8 &= \Sigma X_i - \{f_{\text{BC}}(\Sigma_1^4 X_i, \lambda) - f_{\text{BC}}(\Sigma_5^8 X_i, \lambda)\} \\
&- \{f_{\text{BC}}(\Sigma_5^6 X_i, \lambda) - f_{\text{BC}}(\Sigma_7^8 X_i, \lambda)\} - \{f_{\text{BC}}(X_7, \lambda) - f_{\text{BC}}(X_8, \lambda)\}, \quad (54)
\end{aligned}$$

For the Jacobian we need to differentiate each Y_i by X_j for both $i, j = 1, \ldots, n$. For example,

$$\frac{\partial Y_1}{\partial X_1} = 1 + A + C + G, \quad (55)$$

where $A = (\sum_1^4 X_i)^{\lambda-1}$, $C = (\sum_1^2 X_i)^{\lambda-1}$ and $G = X_1^{\lambda-1}$. Define $B = (\sum_4^8 X_i)^{\lambda-1}$, $D = (\sum_3^4 X_i)^{\lambda-1}$, $E = (\sum_5^6 X_i)^{\lambda-1}$, $F = (\sum_7^8 X_i)^{\lambda-1}$, $H = X_2^{\lambda-1}$, $I = X_3^{\lambda-1}$, $J = X_4^{\lambda-1}$, $K = X_5^{\lambda-1}$, $L = X_6^{\lambda-1}$, $M = X_7^{\lambda-1}$ and $N = X_8^{\lambda-1}$. The Jaco-

bian is

$$\begin{vmatrix}
1+A+C+G & 1+A+C-H & 1+A-D & 1+A-D & 1-B & 1-B & 1-B & 1-B \\
1+A+C-G & 1+A+C+H & 1+A-D & 1+A-D & 1-B & 1-B & 1-B & 1-B \\
1+A-C & 1+A-C & 1+A+D+I & 1+A+D-J & 1-B & 1-B & 1-B & 1-B \\
1+A-C & 1+A-C & 1+A+D-I & 1+A+D+J & 1-B & 1-B & 1-B & 1-B \\
1-A & 1-A & 1-A & 1-A & 1+B+E+K & 1+B+E-L & 1+B-F & 1+B-F \\
1-A & 1-A & 1-A & 1-A & 1+B+E-K & 1+B+E+L & 1+B-F & 1+B-F \\
1-A & 1-A & 1-A & 1-A & 1+B-E & 1+B-E & 1+B+F+M & 1+B+F-N \\
1-A & 1-A & 1-A & 1-A & 1+B-E & 1+B-E & 1+B+F-M & 1+B+F+N
\end{vmatrix}$$

After executing steps 1 and 2 of the proof gives

$$2^4 \begin{vmatrix}
1+A+C & 1+A+C & 1+A-D & 1+A-D & 1-B & 1-B & 1-B & 1-B \\
-G & H & 0 & 0 & 0 & 0 & 0 & 0 \\
1+A-C & 1+A-C & 1+A+D & 1+A+D & 1-B & 1-B & 1-B & 1-B \\
0 & 0 & -I & J & 0 & 0 & 0 & 0 \\
1-A & 1-A & 1-A & 1-A & 1+B+E & 1+B+E & 1+B-F & 1+B-F \\
0 & 0 & 0 & 0 & -K & L & 0 & 0 \\
1-A & 1-A & 1-A & 1-A & 1+B-E & 1+B-E & 1+B+F & 1+B+F \\
0 & 0 & 0 & 0 & 0 & 0 & -M & N
\end{vmatrix}$$

Then execute step 3 to obtain:

$$2^4 2^2 \begin{vmatrix}
1+A & 1+A & 1+A & 1+A & 1-B & 1-B & 1-B & 1-B \\
-G & H & 0 & 0 & 0 & 0 & 0 & 0 \\
-C & -C & D & D & 0 & 0 & 0 & 0 \\
0 & 0 & -I & J & 0 & 0 & 0 & 0 \\
1-A & 1-A & 1-A & 1-A & 1+B & 1+B & 1+B & 1+B \\
0 & 0 & 0 & 0 & -K & L & 0 & 0 \\
0 & 0 & 0 & 0 & -E & -E & F & F \\
0 & 0 & 0 & 0 & 0 & 0 & -M & N
\end{vmatrix}$$

Then the next step (4/5), followed by the first set of column operations gives:

$$2^4 2^2 2 \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -G & H & 0 & 0 & 0 & 0 & 0 & 0 \\ -C & -C & D & D & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & J & 0 & 0 & 0 & 0 \\ -A & -A & -A & -A & B & B & B & B \\ 0 & 0 & 0 & 0 & -K & L & 0 & 0 \\ 0 & 0 & 0 & 0 & -E & -E & F & F \\ 0 & 0 & 0 & 0 & 0 & 0 & -M & N \end{vmatrix} \sim 2^7 \begin{vmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ -G & H+G & 0 & 0 & 0 & 0 & 0 & 0 \\ -C & 0 & D & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & J+I & 0 & 0 & 0 & 0 \\ -A & 0 & -A & 0 & B & 0 & B & 0 \\ 0 & 0 & 0 & 0 & -K & L+K & 0 & 0 \\ 0 & 0 & 0 & 0 & -E & 0 & F & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -M & N+M \end{vmatrix}$$

Then perform the next scale column operation (3rd subtract 1st, and 7th subtract 5th) gives

$$2^7 \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -G & H+G & G & 0 & 0 & 0 & 0 & 0 \\ -C & 0 & D+C & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & J+I & 0 & 0 & 0 & 0 \\ -A & 0 & 0 & 0 & B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -K & L+K & K & 0 \\ 0 & 0 & 0 & 0 & -E & 0 & F+E & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -M & N+M \end{vmatrix}$$

and then the 5th subtract the first gives:

$$2^7 \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -G & H+G & G & 0 & G & 0 & 0 & 0 \\ -C & 0 & D+C & 0 & C & 0 & 0 & 0 \\ 0 & 0 & -I & J+I & 0 & 0 & 0 & 0 \\ -A & 0 & 0 & 0 & A+B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -K & L+K & K & 0 \\ 0 & 0 & 0 & 0 & -E & 0 & F+E & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -M & N+M \end{vmatrix}$$

Now we use Laplace's method to obtain the determinant, pivoting first on the top left element as described in step 7.

$$2^7 \begin{vmatrix} H+G & G & 0 & G & 0 & 0 & 0 \\ 0 & D+C & 0 & C & 0 & 0 & 0 \\ 0 & -I & J+I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A+B & 0 & 0 & 0 \\ 0 & 0 & 0 & -K & L+K & K & 0 \\ 0 & 0 & 0 & -E & 0 & F+E & 0 \\ 0 & 0 & 0 & 0 & 0 & -M & N+M \end{vmatrix}$$

Now pivot on the finest scale diagonals, $N+M$, $L+K$, $J+I$ and $H+G$:

$$2^7(N+M) \begin{vmatrix} H+G & G & 0 & G & 0 & 0 \\ 0 & D+C & 0 & C & 0 & 0 \\ 0 & -I & J+I & 0 & 0 & 0 \\ 0 & 0 & 0 & A+B & 0 & 0 \\ 0 & 0 & 0 & -K & L+K & K \\ 0 & 0 & 0 & -E & 0 & F+E \end{vmatrix}$$

Then:

$$2^7(N+M)(L+K) \begin{vmatrix} H+G & G & 0 & G & 0 \\ 0 & D+C & 0 & C & 0 \\ 0 & -I & J+I & 0 & 0 \\ 0 & 0 & 0 & A+B & 0 \\ 0 & 0 & 0 & -E & F+E \end{vmatrix}$$

Then:

$$2^7(N+M)(L+K)(J+I)\begin{vmatrix} H+G & G & G & 0 \\ 0 & D+C & C & 0 \\ 0 & 0 & A+B & 0 \\ 0 & 0 & -E & F+E \end{vmatrix}$$

Then:

$$2^7(N+M)(L+K)(J+I)(H+G)\begin{vmatrix} D+C & C & 0 \\ 0 & A+B & 0 \\ 0 & -E & F+E \end{vmatrix}$$

Then the next finest: $D+C$ and/or $A+B$ gives

$$2^7(N+M)(L+K)(J+I)(H+G)(D+C)\begin{vmatrix} A+B & 0 \\ -E & F+E \end{vmatrix}$$

which finally gives the result as $2^7(A+B)(C+D)(E+F)(G+H)(I+J)(K+L)(M+N)$ as required.

References

Anscombe, F. J. (1948) The transformation of Poisson, binomial and negative-binomial data., *Biometrika*, **35**, 246–254.

Arnold, B., Balakrishnan, N., and Nagaraja, H. (1992) *A First Course in Order Statistics*, Wiley, New York.

Atkinson, A. C. (1985) *Plots, Transformations and Regression*, Clarendon Press, Oxford.

Bosq, D. (1991) Modelization, nonparametric estimation and prediction for continuous time processes., in G. Roussas, ed., *Nonparametric functional estimation and related topics*, NATO ASI Series, pp. 509–529, NATO.

Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations, *J. R. Statist. Soc.* B, **26**, 211–252.

Burrus, C. S., Gopinath, R. A., and Guo, H. (1997) *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice Hall, Upper Saddle River, NJ.

Cardinali, A. and Nason, G. P. (2010) Costationarity of locally stationary time series, *J. Time Ser. Econom.*, **2**, Article 1.

Coifman, R. R. and Donoho, D. L. (1995) Translation-invariant de-noising, in A. Antoniadis and G. Oppenheim, eds., *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pp. 125–150, Springer-Verlag, New York.

Cole, T. J. and Green, P. J. (1992) Smoothing reference centile curves: the LMS method and penalized likelihood, *Stat. Med.*, **11**, 1305–1319.

Daubechies, I. (1992) *Ten Lectures on Wavelets*, SIAM, Philadelphia.

Donoho, D. L. (1993) Nonlinear wavelet methods of recovery for signals, densities, and spectra from indirect and noisy data, in *Proceedings of Symposia in Applied Mathematics*, volume 47, American Mathematical Society, Providence: RI.

Emeis, S. (2001) Vertical variation of frequency distributions of wind speed in and above the surface layer observed by sodar, *Meteorlogische Zeitschrift*, **10**, 141–149.

Fisz, M. (1955) The limiting distribution of a function of two independent random variables and its statistical application, *Colloquium Mathematicum*, **3**, 138–146.

Fryzlewicz, P. (2008) Data-driven wavelet-Fisz methodology for nonparametric function estimation, *Elec. J. Stat.*, **2**, 863–896.

Fryzlewicz, P. and Delouille, V. (2005) A data-driven Haar-Fisz transform for multiscale variance stabilization, in *Proceedings of the 13th IEEE Workshop on Statistical Signal Processing, Bordeaux*, pp. 539–544.

Fryzlewicz, P. and Nason, G. P. (2004) A Haar-Fisz algorithm for Poisson intensity estimation, *J. Comp. Graph. Stat.*, **13**, 621–638.

Fryzlewicz, P. and Nason, G. P. (2006) Haar-Fisz estimation of evolutionary wavelet spectra, *J. R. Statist. Soc.* B, **68**, 611–634.

Fryzlewicz, P., Delouille, V., and Nason, G. P. (2007) GOES-8 X-ray sensor variance stabilization using the multiscale data-driven Haar-Fisz transform., *J. R. Statist. Soc.* C, **56**, 99–116.

Fryzlewicz, P., Nason, G. P., and von Sachs, R. (2008) A wavelet-Fisz approach to spectrum estimation, *J. Time Ser. Anal.*, **29**, 868–880.

Fryzlewicz, P. Z. (2003) *Wavelet Techniques for Time Series and Poisson Data*, Ph.D. thesis, University of Bristol, U.K.

Geyer, C. (2006) Breakdown point theory notes, www.stat.umn.edu/geyer/5601/notes/break.pdf.

Hettmansperger, T. P. and McKean, J. W. (1998) *Robust Nonparametric Statistical Methods*, Arnold, London.

Huber, P. and Ronchetti, E. (2009) *Robust Statistics*, Wiley, Hoboken, New Jersey, second edition.

Jansen, M. (2006) Multiscale Poisson data smoothing, *J. R. Statist. Soc.* B, **68**, 27–48.

Jansen, M., Nason, G. P., and Silverman, B. W. (2009) Multiscale methods for data on graphs and irregular multidimensional situations, *J. R. Statist. Soc.* B, **71**, 97–126.

Mallat, S. G. (1989) A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Trans. Patt. Anal. and Mach. Intell.*, **11**, 674–693.

Mallat, S. G. (1998) *A Wavelet Tour of Signal Processing*, Academic Press, San Diego.

Motakis, E. S., Nason, G. P., Fryzlewicz, P., and Rutter, G. A. (2006) Variance stabilization and normalization for one-color microarray data using a data-driven multiscale approach, *Bioinformatics*, **22**, 2547–2553.

Moura, M. S. A., Morettin, P. A., Toloi, C. M. C., and Chiann, C. (2012) Transfer function models with time-varying coefficients, *J. Prob. Stat.*, **2012**, article ID: 451076.

Nason, G. P. (2008) *Wavelet Methods in Statistics with R*, Springer, New York.

Nason, G. P. and Bailey, D. (2008) Estimating the intensity of conflict in Iraq, *J. R. Statist. Soc.* A, **171**, 899–914.

Nason, G. P. and Sapatinas, T. (2002) Wavelet packet transfer function modelling of nonstationary time series, *Statistics and Computing*, **12**, 45–56.

Nason, G. P. and Silverman, B. W. (1995) The stationary wavelet transform and some statistical applications, in A. Antoniadis and G. Oppenheim, eds., *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pp. 281–229, Springer-Verlag, New York.

Nason, G. P., von Sachs, R., and Kroisandt, G. (2000) Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum, *J. R. Statist. Soc.* B, **62**, 271–292.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992) *Numerical Recipes in C, the Art of Scientific Computing*, Cambridge University Press, Cambridge.

R Development Core Team (2009) *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Schmidt, T. and Xu, L. (2008) Some limit results on the Haar-Fisz transform for inhomogeneous poisson signals, *J. Anal. App.*, **27**, 483–497.

Sedgewick, R. and Wayne, K. (2011) *Algorithms*, Addison-Wesley, Boston.

Vidakovic, B. (1999) *Statistical Modeling by Wavelets*, Wiley, New York.

Zhang, B., Fadili, M. J., and Starck, J.-L. (2008) Wavelets, ridgelets and curvelets for Poisson noise removal, *IEEE Trans. Im. Proc.*, **17**, 1093–1108.